Manifold® System Release 8.00 User Manual

© 2011

This user manual may be copied and used to create derivative works subject to the requirements of the Georeference Documentation License. You must indemnify all prior authors and organizations and you may not use this manual or any derivative works to document or promote any products other than Manifold products. See the license for details.
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Welcome</td>
<td>1</td>
</tr>
<tr>
<td>How to Install this Documentation</td>
<td>3</td>
</tr>
<tr>
<td>Read Me First</td>
<td>5</td>
</tr>
<tr>
<td>32-bit and 64-bit Manifold Editions</td>
<td>9</td>
</tr>
<tr>
<td>Introduction</td>
<td>17</td>
</tr>
<tr>
<td>Key Ideas in GIS</td>
<td>21</td>
</tr>
<tr>
<td>Key Ideas</td>
<td>32</td>
</tr>
<tr>
<td>Windows</td>
<td>37</td>
</tr>
<tr>
<td>Getting Started</td>
<td>59</td>
</tr>
<tr>
<td>Creating a Project</td>
<td>74</td>
</tr>
<tr>
<td>Components</td>
<td>77</td>
</tr>
<tr>
<td>Drawings, Images and Maps</td>
<td>86</td>
</tr>
<tr>
<td>Working with Maps</td>
<td>89</td>
</tr>
<tr>
<td>Working with Large Images or Surfaces</td>
<td>96</td>
</tr>
<tr>
<td>Selection</td>
<td>101</td>
</tr>
<tr>
<td>Smart Mouse Selection</td>
<td>129</td>
</tr>
<tr>
<td>Formatting</td>
<td>132</td>
</tr>
<tr>
<td>Adding Text Labels</td>
<td>135</td>
</tr>
<tr>
<td>Adding Legends</td>
<td>138</td>
</tr>
<tr>
<td>Editing</td>
<td>147</td>
</tr>
<tr>
<td>Dialog Mode and Visual Tools</td>
<td>149</td>
</tr>
<tr>
<td>Drawings and Databases</td>
<td>162</td>
</tr>
<tr>
<td>Drawings and Tables</td>
<td>168</td>
</tr>
<tr>
<td>Working with Tables</td>
<td>176</td>
</tr>
<tr>
<td>Working with Queries</td>
<td>182</td>
</tr>
<tr>
<td>Finding / Replacing Data</td>
<td>185</td>
</tr>
<tr>
<td>Copy and Paste As</td>
<td>187</td>
</tr>
<tr>
<td>Working with Text Components</td>
<td>192</td>
</tr>
<tr>
<td>Undo / Redo</td>
<td>194</td>
</tr>
<tr>
<td>Linked Views</td>
<td>195</td>
</tr>
<tr>
<td>About Geocoding</td>
<td>203</td>
</tr>
<tr>
<td>Creating Drawings from Geocoded Tables</td>
<td>213</td>
</tr>
<tr>
<td>Using Projections</td>
<td>226</td>
</tr>
<tr>
<td>Projections and Imported Components</td>
<td>230</td>
</tr>
<tr>
<td>How to Print</td>
<td>236</td>
</tr>
<tr>
<td>Printing to Files or Images</td>
<td>248</td>
</tr>
<tr>
<td>Working with GPS Receivers</td>
<td>250</td>
</tr>
<tr>
<td>Key Windows and Dialogs</td>
<td>267</td>
</tr>
<tr>
<td>Security and Access Control</td>
<td>271</td>
</tr>
<tr>
<td>Where to Get Maps</td>
<td>272</td>
</tr>
<tr>
<td>Performance Tips</td>
<td>280</td>
</tr>
<tr>
<td>Memory Requirements</td>
<td>288</td>
</tr>
<tr>
<td>Managing Cache Files</td>
<td>291</td>
</tr>
<tr>
<td>Using the Manifold ODBC Driver</td>
<td>292</td>
</tr>
<tr>
<td>Command Line Options</td>
<td>294</td>
</tr>
<tr>
<td>For Experienced GIS Users</td>
<td>296</td>
</tr>
<tr>
<td>Data Storage Strategies</td>
<td>309</td>
</tr>
<tr>
<td>Spatial DBMS</td>
<td>312</td>
</tr>
<tr>
<td>ESRI Geodatabases</td>
<td>318</td>
</tr>
</tbody>
</table>
Table of Contents

The Data Source Dialog .......................................................... 320
NVIDIA CUDA ............................................................................ 339
Database Installations ................................................................. 349
Database Installations ................................................................. 349
IBM DB2 Express-C Edition ....................................................... 353
Oracle Express Edition .............................................................. 356
SQL Server Express Edition ...................................................... 358

Drawings ..................................................................................... 367

Editing Drawings ...................................................................... 517
Formatting ................................................................................ 432
Hyperlinks ................................................................................. 429
Linked Drawings ..................................................................... 389
Linked Drawings from Geocoded Tables .................................. 391
Multi-User Editing of Linked Drawings .................................... 413

Transform - Boundaries ............................................................. 616
Transform - Attach to / Attach to Self ....................................... 610
Transform Operators - Drawings ............................................ 605
Transform Toolbar - Drawings ................................................. 601

Segmentize ................................................................................ 582
Districts .................................................................................... 575
Dissolve ..................................................................................... 572
Shared Edit ............................................................................... 566

Autocomplete with ALT ............................................................ 558
Editing Drawings ...................................................................... 517
Editing Drawings ...................................................................... 517
Using Cut, Copy and Paste in Drawings .................................. 519
Adding Points, Lines and Areas .............................................. 524
Adding Shapes .......................................................................... 530
Instant Data ............................................................................. 536
Editing Objects .......................................................................... 540
Editing with Snap ..................................................................... 552

Dissolve ..................................................................................... 572
Districts .................................................................................... 575
Orthogonalize .......................................................................... 580
Segmentize ............................................................................... 582
Simplify .................................................................................... 584
Spatial Overlay .......................................................................... 586
Topology Overlay ..................................................................... 593
Traditional Topology Tools ..................................................... 598

Oracle Express Edition ............................................................. 356
IBM DB2 Express-C Edition ....................................................... 353
Database Installations................................................................ 349
<table>
<thead>
<tr>
<th>Transform - Bounding Boxes</th>
<th>620</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transform - Buffers</td>
<td>621</td>
</tr>
<tr>
<td>Transform - Centroids</td>
<td>624</td>
</tr>
<tr>
<td>Transform - Decompose</td>
<td>628</td>
</tr>
<tr>
<td>Transform - Decompose to Convex Parts</td>
<td>629</td>
</tr>
<tr>
<td>Transform - Decompose to Triangles</td>
<td>631</td>
</tr>
<tr>
<td>Transform - Distance Network</td>
<td>633</td>
</tr>
<tr>
<td>Transform - Enclosing Circles</td>
<td>635</td>
</tr>
<tr>
<td>Transform - Enclosing Rectangles</td>
<td>636</td>
</tr>
<tr>
<td>Transform - Explode</td>
<td>638</td>
</tr>
<tr>
<td>Transform - Join Lines</td>
<td>640</td>
</tr>
<tr>
<td>Transform - Intersection Points</td>
<td>642</td>
</tr>
<tr>
<td>Transform - Node Points</td>
<td>643</td>
</tr>
<tr>
<td>Transform - Points</td>
<td>645</td>
</tr>
<tr>
<td>Transform - Clip with (Intersect) / (Subtract)</td>
<td>646</td>
</tr>
<tr>
<td>Transform - Clusters</td>
<td>649</td>
</tr>
<tr>
<td>Transform - Constrained Triangulation</td>
<td>652</td>
</tr>
<tr>
<td>Transform - Convex Hull</td>
<td>657</td>
</tr>
<tr>
<td>Transform - Flip Horizontally</td>
<td>658</td>
</tr>
<tr>
<td>Transform - Flip Vertically</td>
<td>659</td>
</tr>
<tr>
<td>Transform - Gabriel Network</td>
<td>660</td>
</tr>
<tr>
<td>Transform - Intersect Lines</td>
<td>663</td>
</tr>
<tr>
<td>Transform - Move Horizontally / Move Vertically</td>
<td>664</td>
</tr>
<tr>
<td>Transform - Nearest / Farthest Neighbors</td>
<td>666</td>
</tr>
<tr>
<td>Transform - Normalize Metric</td>
<td>667</td>
</tr>
<tr>
<td>Transform - Normalize Topology</td>
<td>670</td>
</tr>
<tr>
<td>Transform - Relative Neighborhood Network</td>
<td>674</td>
</tr>
<tr>
<td>Transform - Remove Duplicates</td>
<td>677</td>
</tr>
<tr>
<td>Transform - Reverse Lines</td>
<td>678</td>
</tr>
<tr>
<td>Transform - Rotate (Drawings)</td>
<td>680</td>
</tr>
<tr>
<td>Transform - Scale / Scale Horizontally / Scale Vertically</td>
<td>681</td>
</tr>
<tr>
<td>Transform - Segments</td>
<td>683</td>
</tr>
<tr>
<td>Transform - Select Adjacent to</td>
<td>684</td>
</tr>
<tr>
<td>Transform - Select Contained / Containing</td>
<td>685</td>
</tr>
<tr>
<td>Transform - Select Euclidean Point Coverage</td>
<td>687</td>
</tr>
<tr>
<td>Transform - Select Intersecting</td>
<td>690</td>
</tr>
<tr>
<td>Transform - Select Shortest Path</td>
<td>691</td>
</tr>
<tr>
<td>Transform - Select Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Transform - Select Touching</td>
<td>696</td>
</tr>
<tr>
<td>Transform - Shape Hull</td>
<td>697</td>
</tr>
<tr>
<td>Transform - Spanning Tree</td>
<td>698</td>
</tr>
<tr>
<td>Transform - Spline</td>
<td>701</td>
</tr>
<tr>
<td>Transform - Split With</td>
<td>703</td>
</tr>
<tr>
<td>Transform - Triangulation</td>
<td>705</td>
</tr>
<tr>
<td>Transform - Union</td>
<td>709</td>
</tr>
<tr>
<td>Transform - Voronoi Operators</td>
<td>710</td>
</tr>
<tr>
<td>Business Tools</td>
<td>712</td>
</tr>
<tr>
<td>Business Tools</td>
<td>712</td>
</tr>
<tr>
<td>Districts (Advanced)</td>
<td>715</td>
</tr>
<tr>
<td>Districts (Visual)</td>
<td>725</td>
</tr>
<tr>
<td>Drive-Time Zones</td>
<td>741</td>
</tr>
<tr>
<td>Optimal Route</td>
<td>748</td>
</tr>
<tr>
<td>Optimal Route (Visual)</td>
<td>750</td>
</tr>
</tbody>
</table>
Table of Contents

Send Email .................................................................................................................................................. 762
Topology Factory ...................................................................................................................................... 767
Images .......................................................................................................................................................... 775
Image Window Menus and Controls ................................................................................................. 783
Image Types ........................................................................................................................................... 789
Images....................................................................................................................................................... 793
Intermediate Levels and Pyramids ..................................................................................................... 799
Linked Images.............................................................................................................................................. 804
Image Libraries ........................................................................................................................................ 809
Image Libraries......................................................................................................................................... 809
Images and Channels ........................................................................................................................... 830
Colors as Hue, Saturation and Brightness ........................................................................................... 836
Invisible Pixels ........................................................................................................................................ 838
Invisible Pixels and Selection ................................................................................................................ 852
RGBa Pixel Transparency ...................................................................................................................... 859
Palettes ..................................................................................................................................................... 865
Images and Surfaces .................................................................................................................................. 872
Linked Images from Image Servers .................................................................................................... 872
Linked Images from Manifold Image Servers ..................................................................................... 872
Linked Images from Manifold IMS Web Sites ..................................................................................... 879
Linked Images from Google Servers .................................................................................................. 881
Linked Images from OGC WMS Servers .............................................................................................. 883
Linked Images from Oracle Servers .................................................................................................. 891
Linked Images from TerraServer .......................................................................................................... 893
Selection in Images ................................................................................................................................... 897
Selection in Images .................................................................................................................................. 897
Selection using Masks .......................................................................................................................... 904
Editing Images ........................................................................................................................................ 906
Editing Images .......................................................................................................................................... 906
Selections and Editing .......................................................................................................................... 909
Copy and Paste in Images ..................................................................................................................... 912
Masks ......................................................................................................................................................... 917
Painting within Channels ....................................................................................................................... 932
Separating Images by Channels .......................................................................................................... 936
Combining Channels into Images ....................................................................................................... 938
Composing Complex Images in Layers ............................................................................................. 942
Editing Palettes ....................................................................................................................................... 947
Image Editing Tools ............................................................................................................................... 948
Image Editing Tools ............................................................................................................................... 948
Paint Bucket Tool ..................................................................................................................................... 955
Gradient Tool ............................................................................................................................................ 963
Transform Toolbar - Images .................................................................................................................. 969
Transform Operators - Images ................................................................................................................ 971
Transform - Add Margin / Crop Margin ............................................................................................... 975
Transform - Auto Contrast .................................................................................................................... 976
Transform - Auto Level ........................................................................................................................ 977
Transform - Crop ....................................................................................................................................... 979
Transform - Desaturate .......................................................................................................................... 981
Transform - Diffuse ................................................................................................................................... 983
Transform - Flip Horizontally / Flip Vertically .................................................................................. 985
Transform - Grayscale ........................................................................................................................... 986
Transform - High Pass ............................................................................................................................. 987
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transform - Median Cross / Median Square</td>
<td>989</td>
</tr>
<tr>
<td>Transform - Rotate</td>
<td>991</td>
</tr>
<tr>
<td>Maps</td>
<td>993</td>
</tr>
<tr>
<td>Maps</td>
<td>993</td>
</tr>
<tr>
<td>Layers</td>
<td>995</td>
</tr>
<tr>
<td>Layers and Commands</td>
<td>1001</td>
</tr>
<tr>
<td>Layer Restrictions</td>
<td>1007</td>
</tr>
<tr>
<td>Layer Opacity</td>
<td>1011</td>
</tr>
<tr>
<td>Repositioning Layers</td>
<td>1019</td>
</tr>
<tr>
<td>Merging Layers</td>
<td>1022</td>
</tr>
<tr>
<td>Zoom Ranges</td>
<td>1023</td>
</tr>
<tr>
<td>Match</td>
<td>1029</td>
</tr>
<tr>
<td>Transfer Selection</td>
<td>1038</td>
</tr>
<tr>
<td>Tracing</td>
<td>1042</td>
</tr>
<tr>
<td>Tracing</td>
<td>1042</td>
</tr>
<tr>
<td>Freehand Tracing</td>
<td>1047</td>
</tr>
<tr>
<td>Tracing Tools</td>
<td>1049</td>
</tr>
<tr>
<td>Labels</td>
<td>1063</td>
</tr>
<tr>
<td>Labels</td>
<td>1063</td>
</tr>
<tr>
<td>Creating Labels from Fields</td>
<td>1078</td>
</tr>
<tr>
<td>Synchronized Labels</td>
<td>1092</td>
</tr>
<tr>
<td>Formatting Labels</td>
<td>1105</td>
</tr>
<tr>
<td>Formatting Individual Labels</td>
<td>1112</td>
</tr>
<tr>
<td>Thematic Formatting and Labels</td>
<td>1118</td>
</tr>
<tr>
<td>Aligning Labels</td>
<td>1124</td>
</tr>
<tr>
<td>Label Display Options</td>
<td>1128</td>
</tr>
<tr>
<td>Highway Shield Labels</td>
<td>1134</td>
</tr>
<tr>
<td>Repositioning Labels</td>
<td>1136</td>
</tr>
<tr>
<td>Rotating Labels</td>
<td>1139</td>
</tr>
<tr>
<td>Editing Labels</td>
<td>1151</td>
</tr>
<tr>
<td>Legends</td>
<td>1163</td>
</tr>
<tr>
<td>Legends</td>
<td>1163</td>
</tr>
<tr>
<td>Legend Text Escape Sequences</td>
<td>1219</td>
</tr>
<tr>
<td>Tables</td>
<td>1221</td>
</tr>
<tr>
<td>Tables</td>
<td>1221</td>
</tr>
<tr>
<td>Importing and Linking Tables</td>
<td>1226</td>
</tr>
<tr>
<td>Table Window Menus and Controls</td>
<td>1233</td>
</tr>
<tr>
<td>Creating New Tables</td>
<td>1239</td>
</tr>
<tr>
<td>Intrinsic Fields in Tables</td>
<td>1244</td>
</tr>
<tr>
<td>Field Types in Tables</td>
<td>1250</td>
</tr>
<tr>
<td>Selection in Tables</td>
<td>1255</td>
</tr>
<tr>
<td>Formatting Columns</td>
<td>1257</td>
</tr>
<tr>
<td>Transfer Rules</td>
<td>1260</td>
</tr>
<tr>
<td>Relations</td>
<td>1268</td>
</tr>
<tr>
<td>Attaching External Tables to Drawings</td>
<td>1272</td>
</tr>
<tr>
<td>Geometry in Tables</td>
<td>1275</td>
</tr>
<tr>
<td>Virtual Tables for Images and Surfaces</td>
<td>1297</td>
</tr>
<tr>
<td>Active Columns</td>
<td>1300</td>
</tr>
<tr>
<td>Finding Data in Tables</td>
<td>1306</td>
</tr>
<tr>
<td>More Like This</td>
<td>1307</td>
</tr>
<tr>
<td>Rank Columns</td>
<td>1310</td>
</tr>
<tr>
<td>Editing Tables</td>
<td>1311</td>
</tr>
<tr>
<td>Editing Data in Tables</td>
<td>1311</td>
</tr>
</tbody>
</table>
# Table of Contents

- Surfaces and Terrains ........................................................................................................................................... 1461
- Charts .................................................................................................................................................................. 1453
- Queries ................................................................................................................................................................. 1353
  - Transform Toolbar - Surfaces .......................................................................................................................... 1518
  - Area Overlays ................................................................................................................................................... 1497
  - Terrains ............................................................................................................................................................. 1483
  - Surface Display Options ................................................................................................................................. 1470
  - Surface Window Menus and Controls ................................................................................................................ 1466
  - Minicharts .......................................................................................................................................................... 1458
  - Charts ............................................................................................................................................................... 1453
  - Introduction to Decision Support .................................................................................................................... 1444
  - Spatial Geocoding with Match .......................................................................................................................... 1436
  - Manifold Geocoding Servers ............................................................................................................................. 1432
  - Geocoding with MapPoint .................................................................................................................................. 1424
  - Geocoding Data Sources .................................................................................................................................. 1421
  - Geocoding Tools ............................................................................................................................................... 1405
  - Geocoding .......................................................................................................................................................... 1403
  - Changing Field Types in Tables ........................................................................................................................ 1332
  - Regular Expressions ......................................................................................................................................... 1324
  - Editing Intrinsic Fields in Tables ...................................................................................................................... 1321
  - Adding or Deleting Fields in Tables ................................................................................................................ 1331
  - Transform Operators - Tables ........................................................................................................................... 1335
  - Transform Operators - Surfaces ......................................................................................................................... 1520
  - Changing Field Types in Tables ........................................................................................................................ 1332
  - Transform - Using Tokens and Text Strings .................................................................................................... 1351
  - Transform - Span Excluding / Including ....................................................................................................... 1352
  - Queries and Geoms ........................................................................................................................................... 1386
  - Queries Using Multiple Tables .......................................................................................................................... 1373
  - Calculations in Queries ..................................................................................................................................... 1375
  - Action Queries .................................................................................................................................................. 1377
  - Parameter Queries .......................................................................................................................................... 1378
  - Selecting Objects with Queries ....................................................................................................................... 1380
  - Queries and Images or Surfaces ...................................................................................................................... 1390
  - Simple Queries ................................................................................................................................................ 1371
  - Queries ............................................................................................................................................................ 1363
- Geocoding ............................................................................................................................................................... 1403
- Charts .................................................................................................................................................................. 1453
- Editing Intrinsic Fields in Tables ...................................................................................................................... 1321
- Regular Expressions ......................................................................................................................................... 1324
- Adding or Deleting Fields in Tables ................................................................................................................ 1331
- Changing Field Types in Tables ........................................................................................................................ 1332
- Transform Toolbar - Tables ............................................................................................................................... 1332
- Transform Toolbar - Surface - Interpolate Operators ........................................................................................ 1523
- Transform Operators - Tables ........................................................................................................................... 1335
- Transform Operators - Surfaces ......................................................................................................................... 1520
- Transform - Surface - Tile Operators ................................................................................................................ 1526
Table of Contents

Surface Tools ................................................................................................................................................................ 1527
Surface Tools ................................................................................................................................................................ 1527
Transforming Surfaces .............................................................................................................................................. 1530
Transform Dialog Functions and Operators .................................................................................................................. 1541
Visible Area .................................................................................................................................................................. 1547
Profiles and Elevations .............................................................................................................................................. 1555
Profiles and Elevations .............................................................................................................................................. 1555
Analysis ........................................................................................................................................................................ 1565
Analysis ........................................................................................................................................................................ 1565
Measurement ............................................................................................................................................................... 1566
Tracker .......................................................................................................................................................................... 1567
ViewBots ...................................................................................................................................................................... 1571
ViewBot Operators ....................................................................................................................................................... 1583
Printing and Layouts ...................................................................................................................................................... 1587
Printing .......................................................................................................................................................................... 1587
Layouts ........................................................................................................................................................................... 1590
Layout Templates .......................................................................................................................................................... 1628
Specifying Views in Layouts ........................................................................................................................................... 1640
Multipage Layouts .......................................................................................................................................................... 1655
Legends, Scale Bars and North Arrows in Layouts .......................................................................................................... 1671
Printing Tables ............................................................................................................................................................... 1673
Import and Export .......................................................................................................................................................... 1679
Import and Export .......................................................................................................................................................... 1679
XML Accessory File Format ........................................................................................................................................... 1682
Import Component .......................................................................................................................................................... 1691
Import Drawing ............................................................................................................................................................... 1691
Importing Drawings .......................................................................................................................................................... 1691
Import Drawing - ADF ....................................................................................................................................................... 1698
Import Drawing - ADO .NET / ODBC / OLE DB ................................................................................................................. 1699
Import Drawing - BNA ....................................................................................................................................................... 1700
Import Drawing - CSV ....................................................................................................................................................... 1701
Import Drawing - DB ....................................................................................................................................................... 1702
Import Drawing - DBF ....................................................................................................................................................... 1703
Import Drawing - DGN ....................................................................................................................................................... 1704
Import Drawing - DLG ....................................................................................................................................................... 1705
Import Drawing - DSN ....................................................................................................................................................... 1706
Import Drawing - DWG ....................................................................................................................................................... 1707
Import Drawing - DXF ....................................................................................................................................................... 1708
Import Drawing - E00 ....................................................................................................................................................... 1715
Import Drawing - GDF ....................................................................................................................................................... 1716
Import Drawing - Geocoding Database ............................................................................................................................. 1717
Import Drawing - Geodatabase ........................................................................................................................................ 1727
Import Drawing - GML ....................................................................................................................................................... 1728
Import Drawing - KML, KMZ ........................................................................................................................................... 1734
Import Drawing - LULC ....................................................................................................................................................... 1735
Import Drawing - ETAK MapBase .................................................................................................................................... 1738
Import Drawing - HTML .................................................................................................................................................... 1739
Import Drawing - MDB ....................................................................................................................................................... 1740
Import Drawing - MFD ....................................................................................................................................................... 1741
Import Drawing - MIF ....................................................................................................................................................... 1742
Import Drawing - MWS ....................................................................................................................................................... 1743
Import Drawing - NTAD .................................................................................................................................................... 1744
Import Drawing - NTF ....................................................................................................................................................... 1745
# Table of Contents

- **Import Drawing** - SHP, Shapefiles .......................................................... 1748
- **Import Drawing** - TAB .............................................................................. 1755
- **Import Drawing** - TAIF ............................................................................ 1757
- **Import Drawing** - TIGER/Line ................................................................. 1758
- **Import Drawing** - UDL ............................................................................ 1760
- **Import Drawing** - VCT ............................................................................ 1761
- **Import Drawing** - VMAP .......................................................................... 1762
- **Import Drawing** - WKx ............................................................................... 1766
- **Import Drawing** - XLS ............................................................................... 1767
- **Import Surface** ......................................................................................... 1784
- **Importing Surfaces** ................................................................................... 1784
- **Import Surface** - ADF ............................................................................... 1789
- **Import Surface** - DEM ............................................................................... 1790
- **Import Surface** - DTED ............................................................................ 1792
- **Import Surface** - HDF EOS ........................................................................ 1793
- **Import Surface** - Northwood GRC / GRD .................................................. 1794
- **Import Surface** - Raw Text ......................................................................... 1795
- **Import Surface** - Raw Binary ....................................................................... 1796
- **Import Surface** - SRTM ............................................................................. 1798
- **Import Surface** - XYZ .................................................................................. 1801
- **Import Table** ............................................................................................... 1801
- **Importing Tables** ....................................................................................... 1801
- **Import Table** - CSV ................................................................................... 1807
- **Link Table** .................................................................................................... 1808
- **Linking Tables** ............................................................................................. 1808
- **Link Drawing** ............................................................................................... 1810
- **Linking Drawings** ....................................................................................... 1810
- **Link Image** ................................................................................................... 1811
- **Linking Images** ........................................................................................... 1811
- **Link Surface** ................................................................................................. 1814
- **Linking Surfaces** ......................................................................................... 1814
- **Export Drawing** .......................................................................................... 1816
- **Exporting Drawings** ................................................................................... 1816
- **Export Drawing** - DXF ............................................................................... 1819
- **Export Drawing** - E00 ................................................................................ 1820
- **Export Drawing** - KML, KMZ ..................................................................... 1821
- **Export Drawing** - MIF ................................................................................ 1825
- **Export Drawing** - Oracle ............................................................................ 1826
- **Export Drawing** - SDTS ............................................................................ 1829
- **Export Drawing** - SHP, Shapefiles ............................................................ 1830
- **Export Image** ............................................................................................... 1832
- **Exporting Images** ....................................................................................... 1832
- **Export Image** - AI, EMF, PS and PDF ......................................................... 1836
- **Export Image** - ECW / JPEG2000 ................................................................. 1838
- **Export Image** - GeoTIFF ............................................................................. 1841
- **Export Image** - KML, KMZ ......................................................................... 1843
- **Export Image** - Oracle ................................................................................ 1845
Table of Contents

- Export Image - PNG ................................................................. 1848
- Export Layout ......................................................................... 1848
- Exporting Layouts ................................................................. 1848
- Export Layout - AI ................................................................. 1850
- Export Layout - EMF ............................................................... 1851
- Export Layout - PDF ............................................................... 1853
- Export Layout - PS ................................................................. 1855
- Export Surface ........................................................................ 1856
- Exporting Surfaces ............................................................... 1856
- Export Table ........................................................................... 1858
- Exporting Tables ................................................................. 1858
- Export Table - ADO.NET / ODBC / OLE DB ......................... 1861
- Export Table - CSV ................................................................. 1862
- Export Table - DB ................................................................. 1864
- Export Table - DBF ................................................................. 1865
- Export Table - MDB ............................................................... 1867
- Export Table - XLS ................................................................. 1869
- Projections ............................................................................ 1871
- Projections ............................................................................ 1871
- Projecting a Map ................................................................. 1881
- Projections Quick Reference .................................................. 1887
- Projections and Legacy Formats ............................................ 1890
- Projections Tutorial .............................................................. 1895
- Coordinates .......................................................................... 1903
- Coordinates Tutorial ............................................................ 1905
- Coordinates in Projected Maps .............................................. 1914
- Projections and Images .......................................................... 1917
- Changing a Component's Projection ........................................ 1923
- Segmentization ..................................................................... 1929
- Georegistration ................................................................. 1932
- Georegistration ..................................................................... 1932
- Georegistering an Image to a Drawing .................................... 1944
- Georegistering an Image to Known Coordinates .................... 1945
- Error Surfaces ...................................................................... 1949
- Managing Control Points ...................................................... 1955
- Manual Georegistration ......................................................... 1962
- Projections Readings ............................................................. 1963
- Projections Readings ............................................................. 1963
- General Projections Concepts .............................................. 1964
- The Earth as an Ellipsoid ....................................................... 1968
- Guide to Selecting Map Projections ....................................... 1975
- Manifold Projections ............................................................ 1980
- Manifold Projections ............................................................ 1980
- Gauss Kruger ....................................................................... 1983
- Gauss Kruger ....................................................................... 1983
- Graphics ............................................................................... 1983
- Graphics Projections ............................................................ 1983
- National Grids ....................................................................... 1984
- National Grids ....................................................................... 1984
- GS50 Projection ..................................................................... 1988
- State Plane Coordinate System ............................................. 1990
- Standard ............................................................................... 1991
- Latitude / Longitude Projection ............................................. 1991
Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spatial DBMS Facilities</td>
<td>2223</td>
</tr>
<tr>
<td>Database Administrator Edition</td>
<td>2199</td>
</tr>
<tr>
<td>Enterprise Edition</td>
<td>2175</td>
</tr>
<tr>
<td>Customization</td>
<td>2123</td>
</tr>
<tr>
<td>Internet Map Server</td>
<td>2075</td>
</tr>
<tr>
<td>Networks</td>
<td>2067</td>
</tr>
<tr>
<td>Not All Maps are Networks</td>
<td>2072</td>
</tr>
<tr>
<td>Map Server Overview</td>
<td>2075</td>
</tr>
<tr>
<td>IMS Config.txt Options</td>
<td>2103</td>
</tr>
<tr>
<td>IMS Queries</td>
<td>2107</td>
</tr>
<tr>
<td>Publishing Multiple Pages</td>
<td>2113</td>
</tr>
<tr>
<td>Optimizing Performance</td>
<td>2116</td>
</tr>
<tr>
<td>Enterprise Edition</td>
<td>2175</td>
</tr>
<tr>
<td>Creating an Enterprise Server</td>
<td>2185</td>
</tr>
<tr>
<td>Working with Enterprise Edition</td>
<td>2188</td>
</tr>
<tr>
<td>Administering Enterprise Servers</td>
<td>2193</td>
</tr>
<tr>
<td>Database Administrator Edition</td>
<td>2196</td>
</tr>
<tr>
<td>Using Administrator Console</td>
<td>2202</td>
</tr>
<tr>
<td>Database Object Projection</td>
<td>2208</td>
</tr>
<tr>
<td>Database Object Properties</td>
<td>2210</td>
</tr>
<tr>
<td>Spatial DBMS Facilities</td>
<td>2223</td>
</tr>
<tr>
<td>Oracle Spatial DBMS Facilities</td>
<td>2230</td>
</tr>
<tr>
<td>Example: Storing a Drawing in Oracle</td>
<td>2239</td>
</tr>
<tr>
<td>Example: Storing an Image in Oracle</td>
<td>2253</td>
</tr>
</tbody>
</table>
Table of Contents

SQL Server Spatial DBMS Facilities ................................................................. 2262
SQL Server Spatial DBMS Facilities ................................................................. 2262
Example: Configuring SQL Server 2008 ......................................................... 2268
Example: Storing a Drawing in SQL Server 2008 .......................................... 2278
Example: Linking a Drawing from SQL Server 2008 ...................................... 2282
Example: Tracing Virtual Earth into SQL Server 2008 .................................. 2300
Example: Storing an Image in SQL Server 2008 .......................................... 2331
Manifold Spatial Extender for SQL Server ....................................................... 2339
Manifold Spatial DBMS Facilities ................................................................. 2340
Manifold Spatial DBMS Facilities ................................................................. 2340
Example: Storing a Drawing in Manifold Spatial DBMS ............................... 2344
Example: Storing an Image in Manifold Spatial DBMS ............................... 2355
Example: Storing a Surface in Manifold Spatial DBMS ............................... 2364
Programming ................................................................................................. 2371
Programming Manifold ................................................................................. 2371
Developing Applications ................................................................................. 2378
Scripts ............................................................................................................. 2380
Scripts ............................................................................................................. 2380
Click Events .................................................................................................. 2388
Update Batching ............................................................................................. 2389
User Interface Scripting ............................................................................... 2391
Script Examples ............................................................................................. 2393
Script Examples ............................................................................................. 2393
Forms .............................................................................................................. 2395
Forms .............................................................................................................. 2395
Form Properties ............................................................................................. 2397
Form Controls ................................................................................................. 2399
Editing Forms ................................................................................................. 2403
Forms and Scripts ........................................................................................... 2413
Frequently Used Controls .............................................................................. 2418
Control - Tools Toolbar .................................................................................. 2418
Control - Check Box ....................................................................................... 2421
Control - Option Button .................................................................................. 2426
Control - Command Button ........................................................................... 2430
Control - Frame ............................................................................................. 2433
Control - Text Box .......................................................................................... 2436
Control - Static Text Box ............................................................................... 2439
Control - List Box ........................................................................................... 2442
Control - Combo Box ...................................................................................... 2445
Control - Horiz / Vert Scroll Bars .................................................................... 2448
Advanced Controls ......................................................................................... 2449
Control - Tools (Advanced) Toolbar ................................................................. 2449
Control - Animation ....................................................................................... 2452
Control - Chart ............................................................................................... 2453
Control - Image List ....................................................................................... 2454
Control - Image Combo Box .......................................................................... 2456
Control - Date / Time Picker .......................................................................... 2458
Control - Month View ..................................................................................... 2461
Control - List View ......................................................................................... 2464
Control - Progress Bar .................................................................................... 2467
Control - Masked Text Box ............................................................................ 2469
Control - Rich Text Box .................................................................................. 2473
Control - Slider .............................................................................................. 2480
Table of Contents

Control - Tree View ................................................................. 2482
Control - UpDown ................................................................. 2484
Control - Tab Strip .............................................................. 2486
Control - Picture Clip .......................................................... 2488
Control - Multimedia Control ................................................ 2489
Control - Status Bar ............................................................ 2491
Control - Tool Bar ............................................................... 2493
Control - Cool Bar .............................................................. 2496
Control - Common Dialog Control ........................................ 2498
Control - System Info Control ............................................... 2503
Control - Insert ActiveX Control ............................................ 2505
Debugger ............................................................................. 2505
Extensions ........................................................................... 2510
Geocoding Server Interface .................................................. 2512
Image Server Interface ....................................................... 2515
Debugger ............................................................................. 2519
Programming Reference ..................................................... 2519
Scripting Reference ............................................................ 2520
Constants .............................................................................. 2521
Border Constants ................................................................. 2521
BufferType Constants .......................................................... 2522
ByteOrder Constants ........................................................... 2523
CameraType Constants ....................................................... 2524
ChannelInterleaving Constants ............................................. 2525
ColumnAlign Constants ....................................................... 2526
ColumnCategory Constants .................................................. 2527
ColumnFlag Constants ......................................................... 2528
ColumnFormatNeg Constants ............................................... 2530
ColumnFormatPos Constants ............................................... 2532
ColumnType Constants ........................................................ 2533
ComponentType Constants .................................................. 2535
ComputationMode Constants ............................................... 2537
ConvertPolicy Constants ..................................................... 2538
ConvertPrompt Constants .................................................... 2539
CoordinateBand Constants .................................................. 2540
CoordinateSystemParameterType Constants ......................... 2541
DSSAtomType Constants ...................................................... 2542
DSSHedge Constants .......................................................... 2543
DSSJunction Constants ....................................................... 2544
FormatType Constants ........................................................ 2545
GeocodeLevel Constants ..................................................... 2546
GeocodeStatus Constants .................................................... 2547
GeomArea, GeomLine and GeomPoint Constants .................... 2548
ImageType Constants .......................................................... 2549
JpegCompression Constants ............................................... 2550
JpegQuality Constants ......................................................... 2551
LabelAlignMulti Constants ................................................... 2552
LabelAlignX Constants ......................................................... 2553
LabelAlignY Constants ........................................................ 2554
LayoutBorder Constants ....................................................... 2555
LayoutPaging Constants ....................................................... 2556
# Table of Contents

- ConverterPropertySet Object .................................................................................................................. 2640
- CoordinateConverter Object .................................................................................................................. 2641
- CoordinateSystem Object ....................................................................................................................... 2642
- CoordinateSystemParameter Object ...................................................................................................... 2644
- CoordinateSystemParameterSet Object .................................................................................................. 2645
- CoordinateSystemSet Object .................................................................................................................. 2646
- DataSource Object .................................................................................................................................. 2647
- DataSourceColumnSet Object .................................................................................................................... 2650
- Datum Object ........................................................................................................................................... 2651
- DatumSet Object ....................................................................................................................................... 2653
- Document Object ..................................................................................................................................... 2654
- DocumentSet Object ............................................................................................................................... 2660
- Drawing Object ......................................................................................................................................... 2661
- DrawingSelectionSet Object ..................................................................................................................... 2668
- DrawingWindow Object ............................................................................................................................ 2669
- DSSAtom Object ....................................................................................................................................... 2672
- DSSAtomSet Object .................................................................................................................................... 2674
- DSSQuery Object ...................................................................................................................................... 2675
- DSSQueryEntry Object ............................................................................................................................... 2676
- Elevation Object ....................................................................................................................................... 2677
- Ellipsoid Object ....................................................................................................................................... 2680
- EllipsoidSet Object .................................................................................................................................... 2681
- EventArgs Object ...................................................................................................................................... 2682
- Export Object ........................................................................................................................................... 2683
- ExportAdoNet Object ................................................................................................................................. 2684
- ExportAdoNetOdbc Object .......................................................................................................................... 2685
- ExportAdoNetOleDb Object ......................................................................................................................... 2686
- ExportAdoNetSqlServer Object .................................................................................................................. 2687
- ExportAdoNetOracle Object ......................................................................................................................... 2687
- ExportAi Object ......................................................................................................................................... 2688
- ExportBil Object ....................................................................................................................................... 2689
- ExportBmp Object ..................................................................................................................................... 2690
- ExportCsv Object ...................................................................................................................................... 2691
- ExportDb Object ........................................................................................................................................ 2692
- ExportDb2 Object ....................................................................................................................................... 2693
- ExportDbf Object ....................................................................................................................................... 2694
- ExportDxf Object ....................................................................................................................................... 2695
- ExportE00 Object ....................................................................................................................................... 2696
- ExportEcw Object ..................................................................................................................................... 2697
- ExportEmf Object ....................................................................................................................................... 2698
- ExportEmfGdi Object ................................................................................................................................. 2699
- ExportEmfGdiPlus Object ............................................................................................................................ 2700
- ExportFlt Object ....................................................................................................................................... 2701
- ExportGif Object ....................................................................................................................................... 2703
- ExportGrd Object ....................................................................................................................................... 2704
- ExportGrdSurfer6 Object ............................................................................................................................ 2705
- ExportGrdSurfer7 Object ............................................................................................................................ 2706
- ExportGrdSurferAsci Object ......................................................................................................................... 2707
- ExportHtml Object ..................................................................................................................................... 2708
- ExportJpeg Object ...................................................................................................................................... 2709
- ExportKml Object ...................................................................................................................................... 2710
- ExportMdb Object ...................................................................................................................................... 2711
- ExportMfd Object ...................................................................................................................................... 2712
- ExportObject Object ................................................................................................................................ 2713
<table>
<thead>
<tr>
<th>Class Name</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ExportMif Object</td>
<td>2714</td>
</tr>
<tr>
<td>ExportOdbc Object</td>
<td>2715</td>
</tr>
<tr>
<td>ExportOleDb Object</td>
<td>2716</td>
</tr>
<tr>
<td>ExportOracle Object</td>
<td>2717</td>
</tr>
<tr>
<td>ExportPdf Object</td>
<td>2718</td>
</tr>
<tr>
<td>ExportPng Object</td>
<td>2720</td>
</tr>
<tr>
<td>ExportPostgreSQL Object</td>
<td>2721</td>
</tr>
<tr>
<td>ExportPs Object</td>
<td>2722</td>
</tr>
<tr>
<td>ExportRawBinary Object</td>
<td>2724</td>
</tr>
<tr>
<td>ExportSdts Object</td>
<td>2725</td>
</tr>
<tr>
<td>ExportShp Object</td>
<td>2726</td>
</tr>
<tr>
<td>ExportSqlServer Object</td>
<td>2727</td>
</tr>
<tr>
<td>ExportTga Object</td>
<td>2728</td>
</tr>
<tr>
<td>ExportTiff Object</td>
<td>2729</td>
</tr>
<tr>
<td>ExportTxt Object</td>
<td>2730</td>
</tr>
<tr>
<td>ExportWk Object</td>
<td>2731</td>
</tr>
<tr>
<td>ExportXls Object</td>
<td>2732</td>
</tr>
<tr>
<td>ExportXyz Object</td>
<td>2733</td>
</tr>
<tr>
<td>Folder Object</td>
<td>2734</td>
</tr>
<tr>
<td>Font Object</td>
<td>2736</td>
</tr>
<tr>
<td>Format Object</td>
<td>2737</td>
</tr>
<tr>
<td>FormatValue Object</td>
<td>2739</td>
</tr>
<tr>
<td>FormatValueSet Object</td>
<td>2742</td>
</tr>
<tr>
<td>GeocodeMatch Object</td>
<td>2743</td>
</tr>
<tr>
<td>GeocodeMatchSet Object</td>
<td>2744</td>
</tr>
<tr>
<td>Geocoder Object</td>
<td>2745</td>
</tr>
<tr>
<td>Geom Object</td>
<td>2746</td>
</tr>
<tr>
<td>GeomSet Object</td>
<td>2747</td>
</tr>
<tr>
<td>History Object</td>
<td>2750</td>
</tr>
<tr>
<td>Image Object</td>
<td>2753</td>
</tr>
<tr>
<td>ImageSelectionSet Object</td>
<td>2754</td>
</tr>
<tr>
<td>Import Object</td>
<td>2759</td>
</tr>
<tr>
<td>ImportAdf Object</td>
<td>2760</td>
</tr>
<tr>
<td>ImportAdoNet Object</td>
<td>2761</td>
</tr>
<tr>
<td>ImportAdoNetOdbc Object</td>
<td>2763</td>
</tr>
<tr>
<td>ImportAdoNetOleDb Object</td>
<td>2764</td>
</tr>
<tr>
<td>ImportAdoNetOracle Object</td>
<td>2765</td>
</tr>
<tr>
<td>ImportAdoNetSqlServer Object</td>
<td>2766</td>
</tr>
<tr>
<td>ImportAdoNetSqlServer Object</td>
<td>2768</td>
</tr>
<tr>
<td>ImportAdrg Object</td>
<td>2770</td>
</tr>
<tr>
<td>ImportAvhr Object</td>
<td>2771</td>
</tr>
<tr>
<td>ImportBill Object</td>
<td>2772</td>
</tr>
<tr>
<td>ImportBmp Object</td>
<td>2773</td>
</tr>
<tr>
<td>ImportBra Object</td>
<td>2774</td>
</tr>
<tr>
<td>ImportCadrgCib Object</td>
<td>2775</td>
</tr>
<tr>
<td>ImportCeosSeaWifs Object</td>
<td>2776</td>
</tr>
<tr>
<td>ImportCsv Object</td>
<td>2777</td>
</tr>
<tr>
<td>ImportDb Object</td>
<td>2779</td>
</tr>
<tr>
<td>ImportDb2 Object</td>
<td>2781</td>
</tr>
<tr>
<td>ImportDbf Object</td>
<td>2783</td>
</tr>
<tr>
<td>ImportDem Object</td>
<td>2784</td>
</tr>
<tr>
<td>ImportDemGlobe Object</td>
<td>2785</td>
</tr>
<tr>
<td>ImportDemGtopo30 Object</td>
<td>2786</td>
</tr>
<tr>
<td>Import Object</td>
<td>Page</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>ImportPng Object</td>
<td>2845</td>
</tr>
<tr>
<td>ImportOracle Object</td>
<td>2841</td>
</tr>
<tr>
<td>ImportOleDb Object</td>
<td>2839</td>
</tr>
<tr>
<td>ImportOdbc Object</td>
<td>2837</td>
</tr>
<tr>
<td>ImportNtf Object</td>
<td>2836</td>
</tr>
<tr>
<td>ImportNtad Object</td>
<td>2835</td>
</tr>
<tr>
<td>ImportNorthwood Object</td>
<td>2834</td>
</tr>
<tr>
<td>ImportNitf Object</td>
<td>2833</td>
</tr>
<tr>
<td>ImportNetCdf Object</td>
<td>2832</td>
</tr>
<tr>
<td>ImportMws Object</td>
<td>2831</td>
</tr>
<tr>
<td>ImportMfd Object</td>
<td>2829</td>
</tr>
<tr>
<td>ImportMapBase Object</td>
<td>2827</td>
</tr>
<tr>
<td>ImportLulcGiras Object</td>
<td>2826</td>
</tr>
<tr>
<td>ImportLulc Object</td>
<td>2825</td>
</tr>
<tr>
<td>ImportKml Object</td>
<td>2823</td>
</tr>
<tr>
<td>ImportGrass Object</td>
<td>2811</td>
</tr>
<tr>
<td>ImportGrd Object</td>
<td>2813</td>
</tr>
<tr>
<td>ImportGrdSurfer Object</td>
<td>2814</td>
</tr>
<tr>
<td>ImportGxf Object</td>
<td>2815</td>
</tr>
<tr>
<td>ImportHdf Object</td>
<td>2816</td>
</tr>
<tr>
<td>ImportHdEos Object</td>
<td>2817</td>
</tr>
<tr>
<td>ImportHdfSeaWifs Object</td>
<td>2819</td>
</tr>
<tr>
<td>ImportHtml Object</td>
<td>2820</td>
</tr>
<tr>
<td>ImportImdisp Object</td>
<td>2821</td>
</tr>
<tr>
<td>ImportLjpeg Object</td>
<td>2822</td>
</tr>
<tr>
<td>ImportKml Object</td>
<td>2823</td>
</tr>
<tr>
<td>ImportLas Object</td>
<td>2824</td>
</tr>
<tr>
<td>ImportLulcGiras Object</td>
<td>2826</td>
</tr>
<tr>
<td>ImportMapBase Object</td>
<td>2827</td>
</tr>
<tr>
<td>ImportMdb Object</td>
<td>2828</td>
</tr>
<tr>
<td>ImportMfd Object</td>
<td>2829</td>
</tr>
<tr>
<td>ImportMif Object</td>
<td>2830</td>
</tr>
<tr>
<td>ImportMws Object</td>
<td>2831</td>
</tr>
<tr>
<td>ImportNetCdf Object</td>
<td>2832</td>
</tr>
<tr>
<td>ImportNif Object</td>
<td>2833</td>
</tr>
<tr>
<td>ImportNorthwood Object</td>
<td>2834</td>
</tr>
<tr>
<td>ImportNtad Object</td>
<td>2835</td>
</tr>
<tr>
<td>ImportNif Object</td>
<td>2836</td>
</tr>
<tr>
<td>ImportOdbc Object</td>
<td>2837</td>
</tr>
<tr>
<td>ImportOleDb Object</td>
<td>2839</td>
</tr>
<tr>
<td>ImportOracle Object</td>
<td>2841</td>
</tr>
<tr>
<td>ImportPcx Object</td>
<td>2843</td>
</tr>
<tr>
<td>ImportPix Object</td>
<td>2844</td>
</tr>
<tr>
<td>ImportPng Object</td>
<td>2845</td>
</tr>
<tr>
<td>ImportPostgresql Object</td>
<td>2846</td>
</tr>
<tr>
<td>ImportPpm Object</td>
<td>2848</td>
</tr>
<tr>
<td>ImportRawAscii Object</td>
<td>2849</td>
</tr>
<tr>
<td>Object Name</td>
<td>Page</td>
</tr>
<tr>
<td>----------------------------</td>
<td>------</td>
</tr>
<tr>
<td>ImportRawBinary Object</td>
<td>2851</td>
</tr>
<tr>
<td>ImportRst Object</td>
<td>2853</td>
</tr>
<tr>
<td>ImportS57 Object</td>
<td>2854</td>
</tr>
<tr>
<td>ImportSdts Object</td>
<td>2855</td>
</tr>
<tr>
<td>ImportSgi Object</td>
<td>2857</td>
</tr>
<tr>
<td>ImportShp Object</td>
<td>2858</td>
</tr>
<tr>
<td>ImportSid Object</td>
<td>2860</td>
</tr>
<tr>
<td>ImportSpot Object</td>
<td>2861</td>
</tr>
<tr>
<td>ImportSqlServer Object</td>
<td>2862</td>
</tr>
<tr>
<td>ImportSrtm Object</td>
<td>2864</td>
</tr>
<tr>
<td>ImportSun Object</td>
<td>2865</td>
</tr>
<tr>
<td>ImportTab Object</td>
<td>2866</td>
</tr>
<tr>
<td>ImportTaf Object</td>
<td>2867</td>
</tr>
<tr>
<td>ImportTga Object</td>
<td>2868</td>
</tr>
<tr>
<td>ImportTiff Object</td>
<td>2869</td>
</tr>
<tr>
<td>ImportTiger Object</td>
<td>2870</td>
</tr>
<tr>
<td>ImportTxt Object</td>
<td>2871</td>
</tr>
<tr>
<td>ImportUdi Object</td>
<td>2872</td>
</tr>
<tr>
<td>ImportVct Object</td>
<td>2873</td>
</tr>
<tr>
<td>ImportVmap Object</td>
<td>2874</td>
</tr>
<tr>
<td>ImportWk Object</td>
<td>2876</td>
</tr>
<tr>
<td>ImportXis Object</td>
<td>2877</td>
</tr>
<tr>
<td>ImportXyz Object</td>
<td>2878</td>
</tr>
<tr>
<td>Label Object</td>
<td>2880</td>
</tr>
<tr>
<td>Labels Object</td>
<td>2883</td>
</tr>
<tr>
<td>LabelSelectionSet Object</td>
<td>2889</td>
</tr>
<tr>
<td>LabelSet Object</td>
<td>2890</td>
</tr>
<tr>
<td>LabelWindow Object</td>
<td>2892</td>
</tr>
<tr>
<td>Layer Object</td>
<td>2895</td>
</tr>
<tr>
<td>LayerSet Object</td>
<td>2896</td>
</tr>
<tr>
<td>Layout Object</td>
<td>2897</td>
</tr>
<tr>
<td>LayoutEntry Object</td>
<td>2901</td>
</tr>
<tr>
<td>LayoutEntrySet Object</td>
<td>2905</td>
</tr>
<tr>
<td>LayoutWindow Object</td>
<td>2906</td>
</tr>
<tr>
<td>LookupValue Object</td>
<td>2908</td>
</tr>
<tr>
<td>Map Object</td>
<td>2909</td>
</tr>
<tr>
<td>MapSelectionSet Object</td>
<td>2912</td>
</tr>
<tr>
<td>MapServer Object</td>
<td>2913</td>
</tr>
<tr>
<td>MapServerOgcWms Object</td>
<td>2918</td>
</tr>
<tr>
<td>MapWindow Object</td>
<td>2919</td>
</tr>
<tr>
<td>Object Object</td>
<td>2922</td>
</tr>
<tr>
<td>ObjectSet Object</td>
<td>2925</td>
</tr>
<tr>
<td>Palette Object</td>
<td>2927</td>
</tr>
<tr>
<td>PaletteSelectionSet Object</td>
<td>2930</td>
</tr>
<tr>
<td>Picture Object</td>
<td>2931</td>
</tr>
<tr>
<td>Pixel Object</td>
<td>2932</td>
</tr>
<tr>
<td>PixelSet Object</td>
<td>2934</td>
</tr>
<tr>
<td>Point Object</td>
<td>2936</td>
</tr>
<tr>
<td>PointSet Object</td>
<td>2938</td>
</tr>
<tr>
<td>Profile Object</td>
<td>2939</td>
</tr>
<tr>
<td>Progress Object</td>
<td>2943</td>
</tr>
<tr>
<td>Query Object</td>
<td>2944</td>
</tr>
<tr>
<td>QueryParameter Object</td>
<td>2948</td>
</tr>
</tbody>
</table>
# Table of Contents

- QueryParameterSet Object .................................................. 2949
- RecentDataSource Object .................................................... 2950
- RecentDataSourceSet Object .................................................. 2951
- Record Object ................................................................. 2952
- RecordSet Object ............................................................... 2954
- Rect Object ........................................................................ 2960
- Relation Object .................................................................... 2962
- RelationSet Object .............................................................. 2963
- RichControl Object ............................................................. 2964
- RichForm Object ................................................................. 2965
- Route ..................................................................................... 2967
- Script Object ......................................................................... 2968
- Surface Object ....................................................................... 2971
- SurfaceSelectionSet Object .................................................... 2976
- Table Object .......................................................................... 2977
- TableSelectionSet Object ....................................................... 2981
- TableWindow Object ............................................................ 2982
- Terrain Object ........................................................................ 2983
- TerrainWindow Object .......................................................... 2988
- Theme Object .......................................................................... 2990
- Topology Object ..................................................................... 2995
- TriangleSet Object ............................................................... 2997
- Unit Object ............................................................................. 2998
- UnitSet Object ......................................................................... 2999
- UserInterface Object ............................................................. 3000
- UserInterfaceControl Object ................................................... 3002
- UserInterfaceControlItem Object .............................................. 3003
- UserInterfaceControlItemSet Object ........................................ 3004
- UserInterfaceControlSet Object ............................................... 3005
- UserInterfaceDialog Object ..................................................... 3006
- UserInterfaceDialogSet Object ................................................ 3007
- Version Object ......................................................................... 3008
- View Object ............................................................................ 3009
- ViewSet Object .......................................................................... 3010
- Window Object ......................................................................... 3011
- WindowSet Object ..................................................................... 3013
- Zone Object ............................................................................... 3014
- Zones Object ........................................................................... 3015
- ZoneSelectionSet Object .......................................................... 3017
- ZoneSet Object .......................................................................... 3018
- ControlAlignmentText Constants ............................................... 3019
- ControlAppearanceConstants ....................................................... 3020
- ControlBorderStyleConstants ....................................................... 3021
- ControlCheckValueConstants .................................................... 3022
- ControlMouseButtonDownConstants ............................................... 3023
- ControlMouseMode Constants .................................................... 3024
- ControlMousePointerConstants .................................................... 3026
- ControlMultiSelectConstants ..................................................... 3028
- ControlOleDragConstants ........................................................... 3029
- ControlOleDropConstants .......................................................... 3030
- ControlScrollBarsConstants .......................................................... 3031
- ControlStyleConstants ............................................................. 3032
- ControlStyleComboConstants ...................................................... 3033
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ControlStyleList Constants</td>
<td>3034</td>
</tr>
<tr>
<td>Combo Control</td>
<td>3037</td>
</tr>
<tr>
<td>Command Control</td>
<td>3040</td>
</tr>
<tr>
<td>Component Control</td>
<td>3043</td>
</tr>
<tr>
<td>ControlMouseEventArgs Object</td>
<td>3046</td>
</tr>
<tr>
<td>ControlTrackEventArgs Object</td>
<td>3047</td>
</tr>
<tr>
<td>Frame Control</td>
<td>3048</td>
</tr>
<tr>
<td>HScroll Control</td>
<td>3050</td>
</tr>
<tr>
<td>List Control</td>
<td>3052</td>
</tr>
<tr>
<td>Option Control</td>
<td>3055</td>
</tr>
<tr>
<td>Text Control</td>
<td>3058</td>
</tr>
<tr>
<td>VScroll Control</td>
<td>3061</td>
</tr>
<tr>
<td>Menus, Dialogs and Controls Reference</td>
<td>3063</td>
</tr>
<tr>
<td>Edit Menu</td>
<td>3063</td>
</tr>
<tr>
<td>File Menu</td>
<td>3063</td>
</tr>
<tr>
<td>File - New / Open / Close</td>
<td>3063</td>
</tr>
<tr>
<td>File - Save / Save As</td>
<td>3064</td>
</tr>
<tr>
<td>File - Create</td>
<td>3065</td>
</tr>
<tr>
<td>File - Export</td>
<td>3066</td>
</tr>
<tr>
<td>File - Import</td>
<td>3068</td>
</tr>
<tr>
<td>File - Import - Component</td>
<td>3069</td>
</tr>
<tr>
<td>File - Import - Component from Server</td>
<td>3070</td>
</tr>
<tr>
<td>File - Link</td>
<td>3071</td>
</tr>
<tr>
<td>File - Page Setup</td>
<td>3073</td>
</tr>
<tr>
<td>File - Print</td>
<td>3074</td>
</tr>
<tr>
<td>File - Exit</td>
<td>3075</td>
</tr>
<tr>
<td>Edit Menu</td>
<td>3075</td>
</tr>
<tr>
<td>Edit - Assign Projection</td>
<td>3075</td>
</tr>
<tr>
<td>Edit - Change Projection</td>
<td>3080</td>
</tr>
<tr>
<td>Edit - Undo / Redo</td>
<td>3089</td>
</tr>
<tr>
<td>Edit - Cut / Copy</td>
<td>3090</td>
</tr>
<tr>
<td>Edit - Paste / Paste Append</td>
<td>3091</td>
</tr>
<tr>
<td>Edit - Delete / Delete All</td>
<td>3092</td>
</tr>
<tr>
<td>Edit - Select All / None / Inverse</td>
<td>3093</td>
</tr>
<tr>
<td>Edit - Select Mode</td>
<td>3095</td>
</tr>
<tr>
<td>Edit - Select Objects</td>
<td>3096</td>
</tr>
<tr>
<td>Edit - Select by Type</td>
<td>3097</td>
</tr>
<tr>
<td>Edit - Modify Selection</td>
<td>3098</td>
</tr>
<tr>
<td>Edit - Save, Load Mask/Channel</td>
<td>3099</td>
</tr>
<tr>
<td>Edit - Find / Find Next</td>
<td>3101</td>
</tr>
<tr>
<td>Edit - Replace</td>
<td>3102</td>
</tr>
<tr>
<td>Edit - Advanced</td>
<td>3103</td>
</tr>
<tr>
<td>Edit - Bookmarks</td>
<td>3105</td>
</tr>
<tr>
<td>Edit - Breakpoints</td>
<td>3107</td>
</tr>
<tr>
<td>Edit - Go / Go To</td>
<td>3109</td>
</tr>
<tr>
<td>Edit - Snap To</td>
<td>3111</td>
</tr>
<tr>
<td>Edit - Shared Edit</td>
<td>3118</td>
</tr>
<tr>
<td>Edit - Instant Data</td>
<td>3125</td>
</tr>
<tr>
<td>Edit - Template</td>
<td>3127</td>
</tr>
<tr>
<td>View Menu</td>
<td>3127</td>
</tr>
<tr>
<td>View - Columns</td>
<td>3127</td>
</tr>
<tr>
<td>View - Selection Filter</td>
<td>3129</td>
</tr>
<tr>
<td>View - Sort</td>
<td>3132</td>
</tr>
</tbody>
</table>
Table of Contents

View - Zoom ................................................................................................................................................... 3135
View - Graticule ............................................................................................................................................. 3137
View - Grid..................................................................................................................................................... 3141
View - Legend............................................................................................................................................... 3144
View - North Arrow...................................................................................................................................... 3145
View - Scale Bar .......................................................................................................................................... 3147
View - Structure.......................................................................................................................................... 3149
View - Full Screen....................................................................................................................................... 3153
View - Refresh Data...................................................................................................................................... 3154
View - Refresh / Autorefresh View ............................................................................................................. 3155
View - Display Options .................................................................................................................................. 3156
Panes ............................................................................................................................................................ 3159
View - Panes .................................................................................................................................................. 3159
View - Panes - Call Stack ............................................................................................................................. 3167
View - Panes - Control Points ....................................................................................................................... 3169
View - Panes - Errors ................................................................................................................................. 3173
View - Panes - GPS Console .......................................................................................................................... 3176
View - Panes - History ............................................................................................................................... 3180
View - Panes - Info ....................................................................................................................................... 3181
View - Panes - Layers .................................................................................................................................... 3185
View - Panes - Notes .................................................................................................................................... 3196
View - Panes - Project ................................................................................................................................... 3197
View - Panes - Review .................................................................................................................................. 3199
View - Panes - Selections ............................................................................................................................... 3208
View - Panes - Tool Properties .................................................................................................................... 3211
View - Panes - Variables ............................................................................................................................... 3213
View - Panes - ViewBots ............................................................................................................................ 3215
View - Panes - Views ................................................................................................................................... 3218
View - Panes - Watches ............................................................................................................................... 3220
View - Panes - World .................................................................................................................................... 3222
Properties ...................................................................................................................................................... 3223
View - Properties .......................................................................................................................................... 3223
View - Properties - Link / Share Status ....................................................................................................... 3225
View - Properties - Precision ....................................................................................................................... 3226
View - Properties - Zooms ........................................................................................................................... 3228
Drawing Menu ............................................................................................................................................... 3231
Drawing - Area of Interest ............................................................................................................................ 3232
Drawing - Color ............................................................................................................................................ 3233
Drawing - Dissolve ....................................................................................................................................... 3237
Drawing - Districts ...................................................................................................................................... 3242
Drawing - Districts (Advanced) .................................................................................................................... 3247
Drawing - Districts (Visual) .......................................................................................................................... 3248
Drawing - Drive-Time Zones ......................................................................................................................... 3249
Drawing - Minichart ...................................................................................................................................... 3250
Drawing - Optimal Route ............................................................................................................................... 3252
Drawing - Optimal Route (Visual) .................................................................................................................. 3253
Drawing - Orthogonalize .............................................................................................................................. 3254
Drawing - Relink / Unlink ............................................................................................................................. 3255
Drawing - Segmentize ................................................................................................................................. 3256
Drawing - Send Email .................................................................................................................................. 3258
Drawing - Simplify ....................................................................................................................................... 3259
Drawing - Spatial Overlay ............................................................................................................................. 3261
Drawing - Topology Overlay ......................................................................................................................... 3267
Table of Contents

Drawing - Transfer Heights ................................................................. 3268
Drawing - Visible Area ...................................................................... 3271
Labels Menu .................................................................................... 3278
Labels - Text .................................................................................. 3278
Labels - Synchronized .................................................................... 3279
Labels - Unlink ................................................................................ 3280
Tools Menu .................................................................................... 3280
Tools - Add-In Manager .................................................................. 3280
Tools - Administrator Console ......................................................... 3282
Tools - Database Console ................................................................. 3285
Tools - Make Image ......................................................................... 3297
Tools - Server Console .................................................................... 3299
Tools - Batch Export ....................................................................... 3301
Tools - Customize ........................................................................... 3304
Tools - Options .............................................................................. 3305
Image Menu ................................................................................... 3317
Image Menu.................................................................................... 3317
Image - Brightness / Contrast .......................................................... 3319
Image - Colorize ............................................................................ 3320
Image - Color Balance .................................................................... 3322
Image - Create Index Drawing ......................................................... 3323
Image - Diffuse .............................................................................. 3330
Image - Hue / Saturation ................................................................. 3332
Image - Dither ................................................................................ 3334
Image - Download ........................................................................ 3335
Image - Equalize ........................................................................... 3337
Image - Filter ................................................................................ 3339
Image - Fluoresce .......................................................................... 3344
Image - Noise ................................................................................ 3346
Image - Gamma ............................................................................. 3347
Image - Gaussian Blur .................................................................... 3348
Image - Invert ................................................................................ 3353
Image - Motion Blur ....................................................................... 3355
Image - Posterize .......................................................................... 3358
Image - Quantize .......................................................................... 3359
Image - Relief ................................................................................ 3362
Image - Relink / Unlink .................................................................. 3364
Image - Simplify ............................................................................ 3366
Image - Threshold .......................................................................... 3368
Image - Threshold Color ................................................................ 3375
Image - Tile .................................................................................... 3381
Image - Resize ................................................................................ 3383
Image - Convert To ......................................................................... 3384
Table Menu .................................................................................... 3386
Table - Add Active Column ............................................................... 3386
Table - Add Rank Column ................................................................. 3387
Table - Address - Geocode ............................................................... 3388
Table - Design ............................................................................. 3391
Table - Relations ........................................................................... 3393
Table - Relink / Unlink .................................................................... 3397
Table - Match ................................................................................ 3398
Script Menu ................................................................................... 3399
Script Menu ................................................................................... 3399
xxi
# Table of Contents

- Script - Compile to DLL ................................................. 3400
- Script - Language ...................................................... 3401
- Script - References ......................................................... 3402
- Surface Menu................................................................. 3402
  - Surface Menu ......................................................... 3402
  - Surface - Invert ..................................................... 3404
  - Surface - Threshold ................................................ 3405
  - Surface - Posterize .................................................. 3407
  - Surface - Filter ......................................................... 3408
  - Surface - Noise ......................................................... 3410
  - Surface - Tile ............................................................ 3411
  - Surface - Transform ................................................ 3412
  - Surface - Quantize .................................................... 3414
  - Surface - Resize ........................................................ 3415
  - Surface - Convert To ................................................ 3416
  - Surface - Contours .................................................... 3417
- Terrain Menu................................................................. 3421
  - Terrain - Surface ..................................................... 3421
  - Terrain - Overlay .................................................... 3426
  - Terrain - Clouds ....................................................... 3433
  - Terrain - Fog ............................................................ 3434
  - Terrain - Lighting .................................................... 3435
  - Terrain - Water ........................................................ 3437
  - Terrain - Snap to Surface .......................................... 3439
- Window Menu ............................................................... 3439
  - Window Menu .......................................................... 3439
  - Window - Windows Dialog ......................................... 3441
- Help Menu........................................................................... 3443
  - Help - Activate ........................................................ 3443
  - Help - Activate Extension .......................................... 3446
  - Help - About ............................................................. 3448
  - Help - Check for Updates ........................................... 3449
  - Help - Contents ........................................................ 3450
  - Help - Index .............................................................. 3451
  - Help - Search ............................................................ 3452
  - Help - Manifold on the Web ......................................... 3453
- Toolbars ........................................................................... 3454
  - Toolbars ................................................................. 3454
  - Alignment Toolbar ................................................... 3457
  - Edit Toolbar ............................................................. 3459
  - Format Toolbar .......................................................... 3470
  - Navigation Toolbar .................................................... 3475
  - Query Toolbar ............................................................ 3476
  - Selection Toolbar ...................................................... 3481
  - Snap Toolbar ............................................................. 3484
  - Tools Toolbar ............................................................ 3487
  - Tools (Advanced) Toolbar .......................................... 3492
  - Tracing Toolbar .......................................................... 3494
  - Transform Toolbar ..................................................... 3495
- Context Menus..................................................................... 3496
  - Context Menus .......................................................... 3496
  - Project Pane ............................................................ 3496
  - Project Pane - Context Menus ..................................... 3496
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTERSECT Operator</td>
<td>3607</td>
</tr>
<tr>
<td>IS Operator</td>
<td>3609</td>
</tr>
<tr>
<td>IS NULL Operator</td>
<td>3611</td>
</tr>
<tr>
<td>LEAVING Clause</td>
<td>3612</td>
</tr>
<tr>
<td>LIKE Operator</td>
<td>3613</td>
</tr>
<tr>
<td>LIKEX Operator</td>
<td>3615</td>
</tr>
<tr>
<td>OPTIONS Clause</td>
<td>3616</td>
</tr>
<tr>
<td>ORDER BY Clause</td>
<td>3619</td>
</tr>
<tr>
<td>OUTER (LEFT, RIGHT, FULL) JOIN Operators</td>
<td>3621</td>
</tr>
<tr>
<td>PARAMETERS Declaration</td>
<td>3623</td>
</tr>
<tr>
<td>POSITION Operator</td>
<td>3625</td>
</tr>
<tr>
<td>SPLIT BY Clause</td>
<td>3626</td>
</tr>
<tr>
<td>SUBSTRING Operator</td>
<td>3628</td>
</tr>
<tr>
<td>TRIM Operator</td>
<td>3629</td>
</tr>
<tr>
<td>UNION Operator</td>
<td>3630</td>
</tr>
<tr>
<td>UNIQUE Operator</td>
<td>3631</td>
</tr>
<tr>
<td>WHERE Clause</td>
<td>3632</td>
</tr>
<tr>
<td>Examples</td>
<td>3633</td>
</tr>
<tr>
<td>My First Map</td>
<td>3635</td>
</tr>
<tr>
<td>Add Points with Instant Data</td>
<td>3652</td>
</tr>
<tr>
<td>Create a Projected US Map</td>
<td>3663</td>
</tr>
<tr>
<td>Using Layers</td>
<td>3677</td>
</tr>
<tr>
<td>Color a Drawing</td>
<td>3687</td>
</tr>
<tr>
<td>Import a Drawing from the Geocoding Database</td>
<td>3693</td>
</tr>
<tr>
<td>Fun with Google Earth</td>
<td>3701</td>
</tr>
<tr>
<td>Display Demographic Data in a Thematic Map</td>
<td>3723</td>
</tr>
<tr>
<td>Selection in Images</td>
<td>3734</td>
</tr>
<tr>
<td>Displaying Data in a Gradient Map</td>
<td>3740</td>
</tr>
<tr>
<td>Turning Layers Off/On by Zoom</td>
<td>3752</td>
</tr>
<tr>
<td>Create a Table and Add Records</td>
<td>3761</td>
</tr>
<tr>
<td>Edit a Table with the Transform Toolbar</td>
<td>3765</td>
</tr>
<tr>
<td>Create a Map from a Geocoded Table</td>
<td>3776</td>
</tr>
<tr>
<td>Create a Geocoded Table from a Map</td>
<td>3788</td>
</tr>
<tr>
<td>Create a Linked Drawing from a Geocoded Table</td>
<td>3792</td>
</tr>
<tr>
<td>Formatting Lines in a Linked Drawing</td>
<td>3804</td>
</tr>
<tr>
<td>Storing Drawings in SQL Server</td>
<td>3815</td>
</tr>
<tr>
<td>A Flashy Demo - Web Queries and KML</td>
<td>3835</td>
</tr>
<tr>
<td>Creating Bordered Lines</td>
<td>3853</td>
</tr>
<tr>
<td>Georegister a Scanned Paper Map</td>
<td>3861</td>
</tr>
<tr>
<td>Intersection Overlays</td>
<td>3874</td>
</tr>
<tr>
<td>Transfer Contour Line Height to Points</td>
<td>3881</td>
</tr>
<tr>
<td>Color Areas by Counts</td>
<td>3884</td>
</tr>
<tr>
<td>Combine a Surface and a Drawing in a Map</td>
<td>3895</td>
</tr>
<tr>
<td>Adding a Legend</td>
<td>3908</td>
</tr>
<tr>
<td>Graticule Labels in a Print Layout</td>
<td>3926</td>
</tr>
<tr>
<td>Align Items in Layouts</td>
<td>3931</td>
</tr>
<tr>
<td>Create a Print Layout from a Table</td>
<td>3940</td>
</tr>
<tr>
<td>Editing a Surface for Visual Effect</td>
<td>3945</td>
</tr>
<tr>
<td>Copy and Paste: Image / Table / Drawing</td>
<td>3952</td>
</tr>
<tr>
<td>Painting into the Alpha Channel</td>
<td>3957</td>
</tr>
<tr>
<td>Create a Circular Feathered Image</td>
<td>3967</td>
</tr>
<tr>
<td>Manually Georegister an Image</td>
<td>3972</td>
</tr>
</tbody>
</table>
# Table of Contents

- Image Effects in Maps ................................................................. 3982
- Create a Globe Image .................................................................. 4000
- Import a Shapefile ........................................................................ 4006
- Import a Projected Shapefile ........................................................ 4015
- Import a Raw Binary File - NLCD ............................................... 4022
- Import a VMAP Level 1 File .......................................................... 4029
- Query Templates ........................................................................... 4033
- Sample Queries ............................................................................. 4036
- Using SQL to Select Map Objects ................................................. 4042
- Rank Columns / Decision Support System .................................... 4044
- Shortest Path over Land ............................................................... 4051
- Extract Last Names using Tokens ............................................... 4063
- Import Landsat Files and Create Composite RGB Image ............ 4073
- Cookie Cutter a Large Image with Transfer Selection ................ 4086
- Download and Mosaic TerraServer Images ................................ 4101
- Active Columns using VBScript ................................................... 4113
- Active Columns using JScript ....................................................... 4117
- Show Area of a Parcel in Acres ..................................................... 4120
- Exporting KML to Google Earth .................................................. 4131
- Other Examples ............................................................................ 4172
- About the Sample Images ............................................................ 4175
- Troubleshooting ........................................................................... 4181
- Troubleshooting ........................................................................... 4181
- Problems with Installation or Configuration ............................... 4183
- Problems when using Windows 95, 98 or Me ............................. 4185
- Problems Importing or Exporting ............................................... 4186
- Problems with Commands or Tools ............................................. 4190
- Problems with Tables ................................................................. 4196
- Problems with Performance ....................................................... 4200
- Problems with Projections ......................................................... 4202
- Problems with the Internet Map Server ....................................... 4204
- Technical Support ......................................................................... 4213
- Technical Support ......................................................................... 4213
- Activation and Licensing ............................................................ 4227
- Activation Keys and Serial Numbers .......................................... 4227
- Installing and Activating a Manifold Extension ........................... 4255
- Runtime Licenses ......................................................................... 4258
- Manifold System License Server ............................................... 4259
- Maintaining Your Manifold License .......................................... 4267
- License ....................................................................................... 4274
- Appendices .................................................................................. 4279
- Appendices .................................................................................. 4279
- Tables .......................................................................................... 4279
- Units ............................................................................................ 4279
- Essays ........................................................................................ 4281
- Essays ........................................................................................ 4281
- User Interface Design .................................................................. 4282
- Terminology in GIS ...................................................................... 4284
- Images can be Inefficient ............................................................ 4288
- GIS and Networking ................................................................. 4291
- What about Ajax? ......................................................................... 4296
- Using RAM and other Machine Resources ................................ 4300
- Public Access to Public Data ....................................................... 4303
# Table of Contents

- Zip Codes are Not Areas ................................................................. 4310
- Just Say No to GIFs ............................................................................. 4312
- Beware of Counterfeit Software ......................................................... 4313
- Help Links ......................................................................................... 4314
- Help Links ......................................................................................... 4314
- Arcs ................................................................................................. 4315
- Polygons ......................................................................................... 4316
- Coverages ......................................................................................... 4317
- Grids ............................................................................................... 4318
- Thematic Mapping ........................................................................... 4319
- Georeferencing .............................................................................. 4320
- Digitization .................................................................................... 4321
- Digitizing Tablets ........................................................................... 4322
- Map Algebra .................................................................................... 4323
- Making Mosaics ............................................................................. 4324
- Rasterize ........................................................................................ 4325
- Slope and Aspect from Elevation Data ............................................ 4326
- Table - Transform Toolbar and Operators ...................................... 4327
- Thiessen Polygons ........................................................................ 4328
- Vectorizing ..................................................................................... 4329
- Warping Images ............................................................................ 4330
- What's New .................................................................................... 4330
- What's New .................................................................................... 4330
- Info ................................................................................................ 4334
- Limitations ..................................................................................... 4334
- Contacting manifold.net ................................................................. 4337
- About Manifold System ................................................................. 4341
- About manifold.net ....................................................................... 4344
- Miscellaneous .............................................................................. 4347
- Index .............................................................................................. 4349
Welcome to Manifold® System. Manifold System has exploded onto the GIS scene with unprecedented quality, performance, and value. Release 8 is a major upgrade to the previous release. It introduces many new capabilities while retaining key methods that will be immediately familiar to users of previous versions of Manifold System.

If you are new to Manifold System please do not be misled by the low price. Manifold System is by far the largest and most sophisticated GIS program ever created. Manifold includes many advanced capabilities beyond those found in ordinary GIS or mapping systems. It includes features similar to those in professional graphics editors such as Adobe Photoshop, CAD systems like AutoCAD, database systems like Access or SQL Server, a full forms-based scripting environment, statistics and mathematics, data mining and charting, a sophisticated Internet Map Server, an optional US street address geocoder and one of the richest mapping packages around. There is so much functionality in Manifold that you should plan on spending a day or two reading the documentation and trying out examples to become aware of the full range of the product.

Read at least the topics in the Introduction chapter and the topics in the Examples chapter and you'll see that everything falls into place pretty quickly. Reading the topics in the suggested order will reward you with rapid expertise. Conversely, failing to read the documentation as suggested will cause unnecessary frustration, as Manifold cannot be learned by trying out menus or by quick skimming of Help topics.

More users new to GIS have learned to do GIS with Manifold than with any other system, so no matter what your level of expertise if you can patiently read through the documentation you too will learn Manifold rapidly and effectively.

Please begin by reviewing the Read Me First topic. This is a very important topic that should be read by all users.

Users of previous Manifold editions should read the What's New topic.

Technical Support

Manifold System licenses do not include free technical support. If you have acquired any technical support incidents, either a free incident as part of a marketing promotion or by purchasing one or more incidents, please take care not to waste those incidents. Free incidents are always for tech support by email using the procedures published on the manifold.net web site.

Should you ever need technical support, please read the Technical Support topic before contacting manifold.net. This will help you make best use of any free support incidents you may have available without wasting incidents. Note that all emails of a technical support nature that are sent to any manifold.net email address will be forwarded to tech support and will use up a support incident.

See the Products page on the manifold.net website for current pricing and availability of technical support products.

Documentation for All Versions

By default, Manifold System uses the web-based version of Help. Your computer must have Internet connection to read Help. This documentation is available in a local version that may be downloaded and installed for use when there is no Internet connection available. Please see the How to Install this Documentation topic for details. That topic also describes how you can modify Help or create your own version of Help.

This documentation is written for all versions of Manifold System and options such as Business Tools, Geocoding Tools and Surface Tools. All editions and options share the same source code and user interface features so that any advance in quality, performance or ease-of-use advances will help users of any Manifold edition. All Manifold System editions are installed using the same installation file, either delivered by download or on a single disc. The serial number you provide when first launching Manifold determines which edition will be configured on startup and which options will be enabled.

Some features described in this Help file will apply only to a specific edition, such as Enterprise Edition. For a handy road-map of features in different Manifold editions, see the About manifold.net topic. If you are not familiar with the features packaged with different Manifold editions, take a moment to review that topic now.

Enterprise users should familiarize themselves with the basic operation of Manifold by reading the topics in the Introduction and Examples sections and then read all of the topics in the Enterprise Edition section.
Options

This documentation includes topics that discuss features installed by optional Manifold products. The Surface Tools and Business Tools optional extensions provide additional commands and capabilities. The Geocoding Tools extension turns on street address geocoding capability within Manifold System and also enables rapid and easy creation of county maps in the US.

From time to time, manifold.net may make available free downloads of add-ins or other extensions to Manifold System. At the present writing the Traditional Topology Tools add-in package and the Manifold Spatial Extender for SQL Server may be downloaded from the manifold.net web site.

If you have not purchased these options or downloaded and installed the Traditional Topology Tools add-in or the Manifold Spatial Extender, you will not be able to use the capabilities described. All topics that depend upon an option will make it clear the option is required.

Users are also strongly advised to take advantage of free downloadable Image Server and (for users who have licensed the Geocoding Tools extension) Geocoding Server modules from the product downloads page. In particular, the use of Image Server modules is a revolution in ease of use and convenience when creating background images or background maps.

Important Configuration Notices

**Very Important:** Before starting Manifold, go to the Windows Control Panel Display dialog Effects tab and make sure that the Show window contents while dragging check box is not checked.

- In Windows XP this option is found in the Control Panel's Appearances and Themes - Display choice under the Appearance tab by pressing the Effects button.
- In Windows XP or Windows 2003, from the Start button open the Control panel and then open the Display dialog. Click on the Appearance tab and then press the Effects button. Uncheck the Show window contents while dragging check box.
- In Windows 2000, Windows ME and Windows 98 open the Control panel and then open the Display dialog. Click on the Effects tab, and uncheck the Show window contents while dragging check box.
- In Windows Vista, open the Control Panel and click Appearance and Personalization. Click Customize colors, then click Open classic appearance properties for more color options and click the Effects button. Uncheck the Show window contents while dragging check box.

Checking this box will greatly slow down the system when displaying complex maps and images because it forces a refresh of the window contents with each minor change in mouse position while dragging the window. Make sure the box is unchecked.

**Very Important:** When working with large projects make sure to set your Windows use of virtual memory appropriately. Some editions of Windows, such as Windows XP x64, will set a maximum pagefile size that is too small by default. Drill down through the Control Panel - System applet's Advanced settings to change the maximum pagefile size to four or five times the largest project you will need to use. For example, if you will be working with 10 GB projects set the pagefile size to 50 GB. See the discussion in the Memory Requirements topic.

More from manifold.net

Manifold® System is another fine product from your friends at manifold.net. See the About manifold.net topic for information on other Manifold family products.

We appreciate your business and your vote of confidence in our products. If you have suggestions for improvement write us at sales@manifold.net. Each new release benefits from thousands of user suggestions to make a better Manifold for you.

If you like Manifold and the value that it provides please tell your friends about Manifold and post your views online. Because manifold.net invests into software development instead of into advertising the company depends upon the satisfaction of the Manifold user community for continued success. Enjoy!
How to Install this Documentation

This Help documentation is available in two forms: a web-based version and a Microsoft compiled help (.chm) version. Other forms may also become available over time as users and others keep growing this documentation.

By default, Manifold System uses the web-based version and has no local documentation built installed. When we click the Help menu the web version will be launched. This helps keep product installation packages smaller for faster downloads, it allows more frequent updating of documentation on the web without having to re-issue new product builds and it allows users, resellers, VARs, software developers and others to create their own versions of documentation, including translations into other languages.

To display the web-based version your computer must be connected to Internet. The web-based version is hosted at a variety of sites worldwide, by default in the georeference.org user community at http://www.georeference.org/doc/manifold.htm.

If we would like a local version of the documentation we can download the manifold.chm file from documentation repositories such as the georeference.org user community, and install the manifold.chm file using these instructions. If a manifold.chm file has been installed, when we click the Help menu the .chm version will be launched instead of the web version.

If you want to create your own version of the documentation, for example, a simplified overview or a translation into a language other than English, you can create your documentation, compile it into a Microsoft compiled help file using any one of the help compilation tools that are widely available, name it manifold.chm and install it using the instructions in this topic. In that case your documentation will be launched by Manifold via the Help menu commands. Your computer does not need to be connected to Internet to read Help if the manifold.chm file has been installed.

Downloading and Installing Manifold.chm

By default, Manifold System installation packages do not contain the manifold.chm compiled help package. This file must be downloaded and installed on your system.

Download the manifold.chm file from the georeference.org user community or other web site and place it into your Manifold installation folder, typically C:\Program Files\Manifold System. In some Windows systems, after downloading the manifold.chm file you may have to allow navigation to the internal web pages to see topics: in Windows Explorer right click on manifold.chm, choose Properties and then click the Unblock button.

If you have installed a 64-bit Manifold version and you would like to have Help available when launching Manifold in 32-bit form, you can also place a copy of the manifold.chm in the 32-bit Program Files installation folder, typically C:\Program Files (x86)\Manifold System. That completes the installation.

If you have installed the manifold.chm file, in addition to launching Help from within Manifold via the Help menu, you can also launch Help directly from Windows by double-clicking on the manifold.chm file. Create a shortcut to that file if you use it frequently. You can then drag and drop the shortcut onto your desktop or drag and drop it into the Windows taskbar Start button to pin it to the Start menu for easy access.

Using Help to Create your Own Help

Many users worldwide have contributed to the creation of Help. Help is unbundled from the Manifold System product so it can be an open, user-generated document with few license restrictions. Help is not part of the licensed Manifold System product but is an independent document that users are encouraged to utilize to create their own versions if they want.

You may make as many copies as you like of Help and give it to as many people as you like. Manifold Help documentation may be freely copied and exchanged, adapted, edited, translated, enhanced, slimmed down otherwise altered and used at no charge consistent with the grant of license for use of Help given on the original websites (such as the repository at georeference.org) that either serve the web-based version or provide downloads of the manifold.chm file. For example, college instructors may want to make workbooks using Help or a manager in charge of an application based on Manifold may want to create a simplified overview for new employees who only need to perform a specific task. Translations into other languages are also popular.
There are a few restrictions on the free use of what is intended to be as open usage as possible: Content from Help may not be used for the documentation of or promotion of products other than Manifold products. If you use content from Help you agree to indemnify prior authors and organizations involved in creating or in providing you the content against any claim or costs that may arise from your use of the content, and you must require anyone to whom you provide any derivative work to also agree to that indemnity. See the grant of license for details.
Read Me First

Follow these three steps to master Manifold:

**Step 1: Finish the Installation**

Make sure you have all files and recommended downloads for using Manifold:

- Manifold is downloaded from the manifold.net website's Product Downloads page. That page always has the most recent version available for download. If you have obtained your Manifold installation files from some other source, please follow the instructions below to make sure you have the latest version.
- Download the help examples files from the Product Downloads page. You’ll need these to learn Manifold.
- Download the image servers and geocoding servers from the Product Downloads page and install them. Most Manifold users make heavy use of image servers.
- If you have licensed the Geocoding Tools extension and you will be working in the United States, download the free geocoding database from the Product Downloads page and install it. This is a very large download but well worth it.

**About Updates:** Service packs and updates for Manifold System fix bugs and provide upgrades until the next major release. If you installed Manifold from some other source other than a fresh download from the Product Downloads page, your installation may not be up to date. Your Manifold System installation is not complete if the latest Service pack or update has not been installed.

If your computer is connected to Internet, Manifold will automatically inform you upon startup if a new update has been issued. Please note that this automatic update checking service cannot work if your network installation prevents Manifold from connecting to the update servers. See the discussion in the Tools - Options topic.

To manually determine if a new update has been issued, use the Help - Check for Updates command. It is not a good idea to turn off automatic update checking.

If Manifold informs you that an update is available, download and install it. Updates play an important role in fixing bugs, improving performance and adding new features.

**Very important:** Before starting Manifold, make sure that your Windows **Show window contents while dragging** check box is not checked. This option is located in the Windows Control Panel Display dialog as follows:

- In Windows XP or Windows 2003, from the Start button open the Control panel and then open the Display dialog. Click on the Appearance tab and then press the Effects button. Uncheck the Show window contents while dragging check box.
- In Windows 2000, Windows ME and Windows 98 open the Control panel and then open the Display dialog. Click on the Effects tab, and uncheck the Show window contents while dragging check box.
- In Windows Vista, open the Control Panel and click Appearance and Personalization. Click Customize colors, then click Open classic appearance properties for more color options and click the Effects button. Uncheck the Show window contents while dragging check box.

Checking this option box will unnecessarily slow down the system when displaying complex maps and images, so make sure this option box is unchecked.

**Very Important:** When working with large projects make sure to set your Windows use of virtual memory appropriately. Some editions of Windows, such as Windows XP x64, will set a maximum pagefile size that is too small by default. Drill down through the Control Panel - System applet's Advanced settings to change the maximum pagefile size to four or five times the largest project you will need to use. For example, if you will be working with 10 GB projects set the pagefile size to 50 GB. See the discussion in the Memory Requirements topic.
Very important: When linking images into a project from certain image servers (such as OGC WMS image servers or from Microsoft TerraServer) Manifold will automatically cache files using the settings in the Tools - Options - File Locations dialog. Read the Managing Cache Files topic and make sure you have chosen your cache file locations and have taken measures to assure you won't run out of disk space by allowing your cached files to grow unreasonably large.

Important: If you are upgrading from an earlier Manifold release, read the What's New topic. Take special note of changes made and transition issues.

Step 2: Invest a Few Days to Assure Success

Manifold is a sophisticated system that rewards users with great power once they master a handful of key techniques. It takes a day or two of reading Help topics and experimenting with Examples to understand enough about Manifold to work at an elementary level.

No matter how smart or experienced you are, the minimum amount of reading required to operate Manifold is to read all the topics in the Introduction chapter followed by reading all of the topics in the Examples chapter.

Even if you have a very simple job to do there is no way to shorten this time or to skip necessary topics. It's like learning to drive a car: you must learn to operate the steering wheel, the brake, the gas pedal and the transmission or you don't go anywhere. Just because you want to drive a short distance or because you are under time pressure doesn't mean you can skip learning about the steering wheel or the brakes.

If you have never used a GIS product before, we suggest not skipping over the Key Ideas in GIS and the Key Ideas topics in the beginning of the Introduction chapter. If you are a GIS expert, Manifold is very different from anything you have used before. Experts can avoid wasting time by also reading the topics suggested below. Programmers should not begin design or coding until they learn Manifold as an interactive package - that will provide the conceptual and practical foundation to use the API effectively.

All readers should read the Introduction and then dive right into Windows and the other topics in the Introduction chapter. All users should also read the topics in the Drawings, Images, Maps, Labels, Tables, Surfaces and Queries chapters. Take a break from straight reading from time to time to work through various Examples topics. Don't skip over Examples topics, since later examples assume you have read previous topics. If you are unsure how to use a particular menu item, topics for all menus and dialogs are organized by menu in the Menus, Dialogs and Controls Reference. Enterprise Edition users should also read the topics in the Enterprise Edition chapter.

Advice from Technical Support: Some topics are so detailed you will be tempted to jump directly into them if they look like what you need to do today under deadline pressure. Resist the temptation. All topics assume you've read and understood the foundation material presented in the Introduction and Examples chapters, so other topics do not repeat necessary, but elementary, steps. Jumping into a topic without reading the introductory material in the Introduction and Examples chapters will most likely waste time and cause needless frustration.

Step 3: Follow Through with Continuing Education

Expand your mastery of Manifold with advanced reading in this Help file. Some very serious users will read the entire documentation. Reading the entire Help file will encounter some repetition. Essential information is often repeated in several topics so that it is available when one parachutes into a topic via a hyperlink. That's OK; the repeated information is so essential it will help to read it more than once.

The fastest, lowest cost way to eliminate difficulties is to use the resources that are immediately available to you. See the Troubleshooting topic for tips for fast resolution of problems.

Other questions? Check the Support page for Manifold System at the main www.manifold.net web site. Resources such as Frequently Asked Questions, Updates, guides to specific procedures, supplemental data or software and more are available online. If you have Manifold running, in the main menu clicking on Help - Manifold on the Web will call up a menu of useful web pages.

If you would like to contact manifold.net with questions of a technical nature regarding Manifold System, please read the Technical Support topic first to avoid wasting any free support incidents you may have available. Please note that any questions of a technical support nature, even what may appear to be simple questions, which are sent to any manifold.net email address are processed in accordance with
the procedures set forth in the Technical Support topic. Don't waste your incidents by sending simple tech support questions to sales. Take a moment to search Help or the free resources and save your incidents for really difficult questions the answers to which cannot be found otherwise.

Manifold users meet online in forums such as the forum which may be accessed via the Community page on the manifold.net home page. Postings in the forum are a good way of meeting other Manifold users online. Please keep in mind that the forum is a user community and is not a communications channel to manifold.net; however, news of a technical nature (such as a new update) will be posted to the forum.

See the Help - Manifold on the Web topic for fast access to the Manifold online community in the forum as well as other resources.

There are several online forums operated by Manifold System user groups in different parts of the world in languages other than English. Reading the discussion in online forums and searching their archives is a great way to get more out of Manifold and to avoid expending technical support incidents on matters that are easily answered by colleagues or have already been discussed in a forum.

Third parties provide training for Manifold System. Many users report that self-paced training products are an easier path to learning than relying exclusively upon reading documentation. See the manifold.net web site for links to third parties offering training, or consult the forum for postings about educational opportunities such as classes offered by private companies or by universities.

A Note for Experienced GIS Users

Manifold is a completely new package that introduces many new methods, user interfaces, terms and concepts not available in traditional GIS software. No matter how much you know about GIS and mapping in general it is critically important to read this documentation in the recommended order to avoid wasted effort.

GIS experts can leverage their expertise by beginning with the For Experienced GIS Users topic and then also reading the topics suggested for GIS newbies, beginning with the Introduction. Please do not skip the introductory topics. Although some material will cover ground already known to experts, the introductory topics will also introduce many important details that are essential to successful Manifold operations. Experts may also enjoy reading User Interface Design and other essays in the Appendix.

GIS experts have often honed their expertise in non-Windows operating systems. Because Manifold heavily utilizes numerous Windows ideas, methods and shortcuts it is essential that you become a fluent Windows power user to be successful with Manifold. The Windows topic will teach you what you need to know.

Users of previous Manifold Releases should check out the What's New topic and then jump back into the flow of introductory topics, beginning with the Introduction.

A Note for Users New to GIS

If you are familiar with Microsoft Windows applications and you have the ability to read documentation carefully in the recommended order you'll do just fine with Manifold.

During the first day or so of learning you'll probably think the learning curve is nearly vertical. It is, initially, but then you'll find that things drop into place very rapidly.

Stick with it. Read the documentation carefully without just skimming it and you will be rewarded with the ability to use an entirely new type of software tool that will let you do amazing things.

Serial Numbers and Activation Keys

Manifold System requires a serial number and an Activation key for permanent installation. This copy protection mechanism allows a much lower cost for Manifold System. To get the most out of your license and to not waste activations it is important that you read the Activation Keys and Serial Numbers topic.
Very Important: Manifold is not licensed for endless usage no matter how many times you need to re-install it. Each serial number can be used for only a limited number of activations. Through sensible use and by taking advantage of low-cost upgrades that may be offered from time to time it is fairly easy for virtually all users to continue using Manifold without running out of activations at a very low cost per year even if they encounter computer failures or upgrade computer hardware. See the Activation Keys and Serial Numbers topic for details.

32-bit and 64-bit Editions

Manifold System is published in 32-bit editions and 64-bit editions. Please read the 32-bit and 64-bit Manifold Editions topic when installing, activating or working with 64-bit editions.

Legal Notes

Throughout this documentation the word "Manifold" by itself is an abbreviation for "Manifold® System" which is the full name of the product and includes a ® registered trademark symbol. Because this documentation is licensed material provided to you under the terms and conditions of the End User License Agreement (EULA), you have already agreed as part of your acceptance of the EULA that "Manifold" is a registered trademark. We use the abbreviation to avoid disrupting the easy reading of the text.

Likewise, the terms "buying" or "purchase" may be used in this documentation as casual language referring to the acquisition of Manifold brand products when in fact any such acquisition is a matter of licensing the products. As set forth in the EULA Manifold brand products are always licensed, and never sold. To provide easy reading for average GIS users more relaxed words such as "purchase" are often used in this documentation instead of more technical terms such as license which, as understandable as they are, may disrupt the easy flow of text for people who are not lawyers. Readers should understand that any relaxed language, slang or colloquialisms used in this documentation do not in any way modify the nature of the EULA nor do they convert the licensing of Manifold into a "purchase."

Let's Go!

… Enough preliminaries - it's time to check out the 32-bit and 64-bit Manifold Editions topic for 64-bit software users and then we can dive straight into the Introduction.
32-bit and 64-bit Manifold Editions

Manifold System products are available in native, 64-bit editions for use in x64 Windows operating systems. 64-bit editions of Manifold System can execute in x64 Windows systems running as 100% 64-bit code, making it possible for Manifold installations to take full advantage of modern 64-bit processors and 64-bit Windows systems.

All Manifold 64-bit capabilities are limited to x64 processor architecture as implemented in AMD and Intel x64 processors, including single core and multi core x64 processors. There is no support for Intel Itanium. All topics in this Help file use terms "64-bit" and "x64" interchangeably.

Many Manifold users have adopted 64-bit hardware and 64-bit Windows while other Manifold users have continued to run 32-bit processors and 32-bit Windows. To support both types of users Manifold System is available both in 32-bit products and 64-bit products. 64-bit versions of Manifold have "x64" in their product name. For example, Enterprise Edition may be licensed as either Manifold Enterprise or, for a slightly higher fee, Manifold Enterprise x64.

- A 32-bit Manifold download can be installed into either a 32-bit or a 64-bit Windows system but it can run only in 32-bit mode in either case. A 32-bit Manifold product can run in 32-bit Windows systems in 32-bit mode, or it can run in 64-bit Windows systems in 32-bit mode using the ability of 64-bit Windows to execute legacy 32-bit products in 32-bit mode.

- A 64-bit Manifold download can be installed only into a 64-bit Windows system. When installed in 64-bit Windows systems it can run either in 32-bit mode or in 64-bit mode.

- Licensing a Manifold product provides a serial number that authorizes the associated product. Licensing a 64-bit product provides a 64-bit serial number. Licensing a 32-bit product provides a 32-bit serial number. A 64-bit serial number can authorize either a 64-bit download or a 32-bit download. A 32-bit serial number can authorize only a 32-bit download.

Although most Manifold System editions may be licensed either as 32-bit products or as 64-bit products, some products are provided only as 64-bit products. 64-bit products can be used in either 32-bit or 64-bit Windows.

- Personal, Professional, Enterprise, Universal, Database Administrator and Ultimate editions are published in both 32-bit and 64-bit x64 versions. The x64 versions require a slightly higher licensing fee.
- License Server edition is available only in an x64 version and supports use of both 32-bit and 64-bit Manifold versions as floating licenses. This allows the greatest possible flexibility for use of floating licenses, with usage enabled regardless of whether the target machines happen to be 32-bit or 64-bit Windows installations.

- Professional Runtime and Universal Runtime editions are published only as 64-bit editions at the same low price. 32-bit programs using the Manifold object model will use 32-bit Manifold modules and 64-bit programs will use 64-bit Manifold modules, automatically. Installing a Manifold runtime license on a 64-bit operating system will automatically allow the use of Manifold from both 32-bit and 64-bit programs or web sites, automatically. Of course, the x64 Manifold runtime editions may be installed and used on 32-bit Windows using the 32-bit Manifold installation.

- The Business Tools, Geocoding Tools and Surface Tools extensions (as well as the combined Option Pack bundle) are published only as 64-bit editions and will run either in 32-bit or in 64-bit mode depending on whether the host Manifold System license is running in 32-bit or 64-bit mode. For example, if Surface Tools has been installed on a 64-bit Windows system together with Manifold Enterprise x64 and Manifold is launched in 32-bit mode, then the Surface Tools functionality within that package will also run in 32-bit mode. If Manifold is launched in 64-bit mode, the Surface Tools functionality will also run in 64-bit mode. If a 32-bit version of Manifold has been installed then the extensions will always run in 32-bit mode with that system. If that 32-bit version of Manifold System is upgraded to a 64-bit version, there is no need to upgrade the extensions, as they will simply switch automatically to running in 64-bits when the new 64-bit Manifold package is installed.

In a nutshell, all non-interactive Manifold editions such as License Server or runtime editions and all Manifold extensions always support running in both 32-bit and 64-bit modes, while regular Manifold editions may or may not support running in 64-bit mode depending on the license that has been acquired. A 64-bit license must be acquired to enable operation of regular Manifold editions in 64-bit mode.

Checking Status of Serial Numbers

To check the status of any particular serial number, visit the Support page on the www.manifold.net web site to find a link to the Serial Number Status page. The status page may be used to find out if a particular serial number is a 32-bit or 64-bit serial number, whether it is still active or has been revoked (as might occur if it has
been traded in for an upgrade), what product it authorizes and how many Activation keys are available. This same page may also be used to check the status of a technical support token.

**A reminder:** Any use of a serial number, such as checking status via the Serial Number Status page, should be done using Copy and Paste from the original serial number email sent out by manifold.net. Do not use the "masked" version of the serial number that may be displayed in the Help - About dialog which ends in a series of X characters. The masked version displayed in Help - About has had the final characters altered with a series of X characters so that someone who has physical access to your computer cannot steal your Manifold serial number. The masked version displays enough of the serial number so you can determine for your internal record keeping which serial number you used on a particular machine, but not enough of the serial number for someone to be able to steal it and use it to obtain Activation keys or other wise use it.

**64-bit Requirements**

Only a 64-bit Manifold edition can run in 64-bit mode on 64-bit x64 processors running 64-bit Windows editions. Manifold x64 is not supported for Itanium or other non-x64 bit architectures. To run Manifold in 64-bit mode you must do the following:

- Use an x64 processor, such as AMD Athlon64x2 or Intel EM64T.
- Use an x64 edition of Windows, such as Windows XP x64 or Windows Server 2003 x64.
- Install the x64 edition of .NET Framework 2.0.
- Install an x64 edition of Manifold.
- Activate the x64 edition of Manifold using a 64-bit Manifold serial number.

Manifold download pages provide both 32-bit and 64-bit versions of the Manifold installation package. A Manifold System serial number issued for a 64-bit edition of Manifold can activate either a 32-bit installation or a 64-bit installation. A Manifold System serial number issued for a 32-bit edition of Manifold can activate only a 32-bit installation.

A 64-bit edition of Manifold can only be installed if you have x64 Windows running on an x64 processor. If you attempt to install a 64-bit edition of Manifold on a 32-bit Windows edition you will get an error message. That error message is a sure sign your computer system is not set up to run 64-bit software.

**Is My Computer a 64-bit Computer?**

Users sometimes are not sure if their computer system has a 64-bit processor or is running 64-bit Windows. This can be confusing because some manufacturers don't make it clear whether the processor in use is a 32-bit or a 64-bit processor. A further point of confusion is that some people have installed 32-bit Windows even though their processor is 64-bit. It doesn't matter if you have a 64-bit processor if you are running 32-bit Windows: when you do that, as far as Windows is concerned you are running a 32-bit processor and cannot run 64-bit applications.

If you are not sure whether your system meets the requirements for running 64-bit Manifold, the easiest way to find out is to check what version of Windows you are running. Because x64 versions of Windows will install and run only on x64 processors, if you are running x64 Windows you know you can run 64-bit Manifold x64 as well.

Note that because x64 processors can run 32-bit programs, it is possible to install 32-bit Windows on a 64-bit x64 processor. In that case, it doesn't matter that your processor is 64-bits, because Windows will insist on treating it as a 32-bit processor and you will be unable to install or run 64-bit software like Manifold x64.

To determine what version of Windows you have installed, launch the Windows Control Panel - System applet. The version of Windows in use will be cited in the General tab of that applet. The processor in use will also be cited. If an x64 version of Windows is installed that will be clearly marked with an "x64" designation. If the version of Windows is not clearly marked with an "x64" designation then it is 32-bit Windows.

If you have an x64 version of Windows running you can proceed with x64 Manifold installation. If you have 32-bit Windows running you cannot install or run 64-bit Manifold.

It is possible, although unlikely, that if you have 32-bit Windows it has been installed on a 64-bit processor. You can tell if this is the case by noting the processor reported by the System applet. This might not be so easy to determine, as some manufacturers do not use clear processor names.
AMD uses a refreshingly clear and straightforward naming scheme that usually includes "64" in the names of 64-bit AMD processors. Intel in contrast uses a bizarre naming scheme that features confusingly similar names for radically different processors. For example, as of this writing the Intel "Core Duo" processor is a 32-bit processor while the Intel "Core 2 Duo" processor is a 64-bit processor.

If you can't tell from the name of the processor reported by the System applet whether or not it is a 64-bit processor, contact your hardware vendor and ask them or use any good Internet search engine to research the matter on websites providing information on computer hardware and processors.

If you are running a 64-bit processor but someone nonetheless has installed 32-bit Windows you might be able to install 64-bit Windows on that computer to enable you to use your processor as a 64-bit processor. For information on such upgrades contact your system vendor and/or Microsoft.

Important: These instructions for using the Windows Control Panel - System applet are basic Windows instructions that apply to typical Windows installations. Your installation may be slightly different. If so, please review your Microsoft documentation to determine how to tell what version of Windows you are running.

Manifold.net is not a hardware manufacturer and cannot assist you in determining what hardware you have installed. Manifold.net cannot advise you on your hardware configuration or help you determine whether or not the computer hardware you have is running Windows x64 or be upgraded to Windows x64. Please contact your system vendor and/or Microsoft for such assistance.

32-bit and 64-bit Manifold Installation Packages

There are two separate installation packages for Manifold:

- **32-bit Manifold installation** - This installation package installs 32-bit Manifold. The 32-bit Manifold installation package may be installed on either 32-bit or 64-bit Windows systems. One program shortcut will be created named **Manifold System**. When installed on a 64-bit Windows system a 32-bit Manifold installation will run in 32-bit mode using the capability of 64-bit Windows editions to run legacy 32-bit programs.

- **64-bit Manifold installation** - This installation package installs a Manifold package that can run either in 32-bit mode or in 64-bit mode. The 64-bit Manifold installation package may be installed only on 64-bit Windows systems. Two program shortcuts will be created named **Manifold System (32-bit)** and **Manifold System (64-bit)**. Launching Manifold using the **Manifold System (32-bit)** shortcut will launch Manifold in 32-bit mode. Launching Manifold using the **Manifold System (64-bit)** shortcut will launch Manifold in 64-bit mode. Both the 32-bit and 64-bit shortcuts use the same registry entries so that options set by users will be the same when switching back and forth between launching Manifold in either 32-bit or 64-bit mode.

The 64-bit Manifold installation is a mixed package that installs both 32-bit and 64-bit Manifold modules. For the sake of brevity, it is referred to as "64-bit" Manifold. Installing a 64-bit version of Manifold will allow using Manifold in both 32-bit and 64-bit modes on the same system, simultaneously. When running in 64-bit mode Manifold is executing entirely as 64-bit code.

Activation

Manifold serial numbers can authorize either 32-bit or 64-bit licenses. For the sake of brevity, a serial number that authorizes a 32-bit Manifold license is called a **32-bit serial number**, while a serial number that authorizes a 64-bit Manifold license is called **64-bit serial number**.

A 32-bit serial number will only allow Manifold to function in 32-bit mode. A 64-bit serial number will allow Manifold to function in 32-bit as well as 64-bit mode. Any serial number can be used with any installation package on any operating system. It is perfectly OK to use a 64-bit serial number to activate Manifold on a 32-bit operating system.

A 32-bit serial number will only allow Manifold to function in 32-bit mode. A 64-bit serial number will allow Manifold to function in 32-bit as well as 64-bit mode. Any serial number can be used with any installation package on any operating system, but the results will be guided both by the capabilities of the operating system (it must be a 64-bit operating system to allow Manifold to function in 64-bit mode) and by the nature of the serial number used (a 64-bit serial number must be used to enable 64-bit Manifold operation). For example, it is perfectly OK to use a 64-bit serial number to activate Manifold on a 32-bit operating system, but in that case because the operating system is a 32-bit operating system Manifold will run in 32-bit mode even though the serial number used is a 64-bit serial number.
To consider another example, suppose we jumped the gun a bit on computer procurement and bought an Athlon 64 hardware system but we have been running it using 32-bit Windows. We expect to upgrade to 64-bit Windows but we would like to wait for an Intel Core 2 Duo system to become available before upgrading to 64-bit Windows. However, we would like to license 64-bit Manifold right now so that we have a 64-bit license when we are ready to install 64-bit Windows.

In that case we can immediately license 64-bit Manifold and we can install and activate a 32-bit Manifold installation on our 32-bit Windows systems. When we upgrade to our new 64-bit hardware and new 64-bit Windows system we can use the same 64-bit Manifold serial number with the 64-bit Manifold installation package.

**Note:** Extensions are always issued with 64-bit serial numbers

### Advantages of 64-bit Editions

The main advantages of 64-bit Manifold is that 64-bit processors will always work in 64-bit mode. This is often faster than using a 64-bit processor in 32-bit mode and it allows the processor to use 64-bit wide memory fetches for higher bandwidth to memory as well as a larger, 64-bit memory addressing space. This means that the processor can use far more memory and use that memory more efficiently.

When running in 32-bit Windows, a Manifold process can use no more than 2 GB of memory, the 32-bit Windows process space limit. Because of system and application overhead, in actual practice individual tasks within Manifold (such as running a particular transform operator) could have no more than 1 GB of memory to work with in 32-bit Windows. 64-bit Manifold editions running in x64 Windows can use up to 16 Terabytes of process space, the maximum allowed by x64 Windows.

The ability to use more memory and to more efficiently use memory in 64-bit systems ripples throughout the internal architecture of Manifold System and also provides opportunities for Manifold to leverage enhanced capabilities in 64-bit Windows. For example, many data structures are used throughout the system for stacks, buffers and other needs, and these data structures can be made both larger and implemented more efficiently in 64 bits. Windows itself in 64-bit editions has immensely greater capabilities for the capacity of important architectural components such as virtual memory, paging file size, hyperspace, paged and non-paged pools, system cache and system Page Table Entries (PTEs).

The best way to take advantage of such 64-bit capabilities is to install as much RAM as possible in your computer, ideally at least 4 gigabytes. RAM is cheaper than ever, so load your machine up with as much RAM as you can install, and consider motherboards with larger RAM capacities when procuring new systems. RAM has become so inexpensive that thinking in terms of 8 or 12 gigabytes of RAM for a 64-bit system is not unusual. It is also important to increase the size of the Windows pagefile to give Manifold as well as other applications more virtual memory. If a machine is hobbled by insufficient RAM, such as having only a gigabyte of RAM, then the potential advantages of 64-bit operation will not be realized. It makes little sense to go to 64-bit processors and software and not install plenty of RAM.

The most popular 64-bit processors from AMD and Intel are also multi-core, having two or four processor cores in each chip. Because Manifold can take advantage of multiple cores in many key functions and because having extra cores can offload background Windows and other applications processes, it is wise when procuring 64-bit computer systems to also take choose multi-core processors. Prices are so inexpensive for such processors that it makes sense to use them.

The advantages of 64-bit operation are normally seen in applications large enough for the differences between 64-bit and 32-bit operation to be noticeable. For example, clicking open the Help - About dialog will appear just as fast in 32-bit systems as it is in 64-bit systems, and opening a small drawing that takes up only a few megabytes of storage space will likewise seem fast in either environment. In contrast, many larger jobs or computationally intensive analytic tasks (which end up using lots of memory for intermediate algorithmic operations) can operate dramatically faster in 64-bit systems.

However, even in smaller applications or in computers with only 2 GB of memory a 64-bit Manifold system will be faster than a 32-bit system due to the many application and system efficiencies available when working with a 64-bit application in a 64-bit Windows edition. Redisplay of larger drawings, for example, is usually twice as fast on a 64-bit system than on a 32-bit system.

Small, but significant effects add up in hundreds of places to make 64-bit operation more responsive and pleasant than 32-bit operation. For example, 32-bit systems are limited to 16 megabytes of "last rendered" cache (allowing the Back and Forward display buttons to redisplay the contents of a window faster) to store approximately 16 views in cache while 64-bit systems use a full 64 megabytes of applications cache to save
approximately 64 views. This means that when very actively combining use of various pan, zoom-to-fit, Back and Forward commands a 64-bit system will need to re-render large drawings less frequently than the 32-bit system, because the 64-bit system will have a better chance of finding a previously rendered display in the cache.

Limitations of 64-bit Editions

64-bit Windows editions do not include all features included within 32-bit Windows editions, so those Manifold features that depend upon such missing features will not be available when running in 64-bit mode. In general, such Windows features are older Windows features that have been replaced by more modern, more efficient capabilities. Limitations when running Manifold in 64-bit mode include:

- No common ActiveX controls for forms. The Manifold ActiveX control works OK and may be used by programmers.
- No debugger for scripts written in VBScript or JScript.
- No Microsoft Jet DBMS engine, thus no export, import or linking of DB, HTML, MDB, XLS or WKx format files. This includes no access to the MDB parts of Manifold MFD and MapInfo TAB imports. The workaround for importing or exporting such files is to launch Manifold in 32-bit mode by using the Manifold System (32-bit) shortcut, perform the export from or import into a .map project file and then re-launch Manifold in 64-bit mode using the Manifold System (64-bit) shortcut. Linking is more complex: the data must be in some format usable in 64-bit mode within 64-bit Windows systems, such as a SQL Server database. Alternatively, the data can be kept within a Manifold .map project file and linked using the Manifold ODBC driver.

Programmers working in 64-bit environments should move to .NET and .NET languages. Lack of immediate access to "Office" file formats like MDB and XLS is highly annoying but can be worked around by temporarily switching to 32-bit mode. The main hit will be to people who are storing data in Excel XLS spreadsheets or Access DBMS MDB files. But that's a very backward thing to do in 64-bit environments: far better to use a real DBMS engine like SQL Server Express, which provides twice the storage capacity and far, far greater reliability.

What about the Microsoft Access Database Engine 2010 Redistributable?

In 2010 Microsoft issued the Microsoft Access Database Engine 2010 Redistributable package intended to help 64-bit applications connect to Access and Excel in 64-bit Windows. The package is great, a reasonable 64-bit version of much of what is in Jet, but it is limited in that it does not work well in mixed environments typical of people who have been living with 32-bit Office versions in a 64-bit Windows system.

For example, suppose you have been using 64-bit Windows systems for the many years (seven or more) they have been available. Because Microsoft did not make 64-bit versions of Office applications until very recently you probably have 32-bit Office versions installed in your 64-bit Windows system. Try to install the 2010 redistributable package and it won't install: it tells you that first you must remove your 32-bit installation of Office Products. Sure.

If you uninstall your existing Office applications, acquire 64-bit versions of those Office applications and then install the 2010 redistributable package, a 64-bit Manifold edition will be able to use it to import from .mdb files and .xls files and it will be able to link to these Access and Excel files. But it still won't be the same as real Jet. In fairness to Microsoft, the download page for the 2010 redistributable makes it clear it is not intended as a general replacement for Jet and within the published limitations the package works flawlessly.

However, most users are unwilling to uninstall their existing Office applications and to buy new 64-bit versions just to open an .mdb or .xls file. If you need to do this in a Manifold context we think a much simpler situation is to just launch Manifold as a 32-bit application and connect directly. No muss, no fuss.

For those users who are using 64-bit Excel and need to dynamically link to Excel .xls in scenarios like that in the A Flashy Demo - Web Queries and KML topic, the new redistributable is great. Search the Microsoft downloads website for "Microsoft Access Database Engine 2010 Redistributable" to find the download page.

Launching Help

When an x64 Manifold license is installed on x64 Windows, both 64-bit and 32-bit Manifold applications are installed. The 64-bit application will be launched by default, but the 32-bit application is available for special purposes as noted above. Manifold Help is connected to the Manifold 64-bit application, so Help is not available from the Help menu when the 32-bit Manifold application is running.
Manifold Help can always be launched directly from Windows. Press the Windows Start button in the Windows taskbar, and then choose All Programs, Manifold System and then Manifold System Help. It is often convenient to launch Help directly from Windows. Some Manifold users will launch Help in this way and keep that Help session available on their Windows taskbar for easy reference.

**Advice to Users**

Running 64-bit software is such a huge performance and reliability gain over 32-bit operation that users who have 64-bit hardware should make it a point to run 64-bit Windows and 64-bit Manifold. Don't hesitate - go 64 bits. It is almost certain that any reasonably newer system will already consist of 64-bit hardware, even if the processor name is some confusing name that does not make it clear the processor is indeed a 64-bit processor.

Pushing against this common sense approach to using the most reliable, most cost-effective and fastest systems are computer system vendors who sell 64-bit processors in systems loaded with 32-bit Windows. For example, most notebook computers use 64-bit processors but very few are available (as of this writing) with 64-bit Windows. Users are often shocked to discover that they have purchased a 64-bit processor but that the computer delivered to them is running that processor only in 32-bit mode because 32-bit Windows has been installed. Users are often especially shocked when they learn there is no possibility of installing 64-bit Windows on the computer because the manufacturer will not provide 64-bit drivers for their proprietary hardware.

Computer systems manufacturers will often use 64-bit chips because that is where the volume deals are available from the microprocessor vendors and that is the direction in which all modern microprocessor manufacturing is headed. But to save a bit of time and rush out a new model a systems vendor might be tempted to skip the small amount of effort to provide 64-bit drivers for their proprietary hardware so that 64-bit Windows can be used on the 64-bit hardware they sell. Perhaps they think their customers will neither notice nor care, which could well be true for many consumers.

This can be frustrating because major systems vendors who are most guilty of such practices do create very beautiful and desirable hardware. Some modern notebook computers are indeed exquisite machines as hardware even if their manufacturers are completely careless about software. That means one must often choose from second tier vendors and buy a heavier notebook than desired to get 64-bit Windows support on a portable computer.

In the case of desktop machines, it can be very frustrating to try to procure machines loaded with 64-bit Windows as well. Some vendors may be unaware that 64-bit Windows is for everyone, and may seem to want to sell it only through their "enterprise" sales divisions to server customers, as if the individual customer to whom they would sell a 64-bit processor-based machine would not also want to use all 64-bits of their processor!

One way to avoid such problems is to buy hardware from vendors who cater to gamers, since they tend to know all about the need to provide 64-bit Windows for performance, even on a desktop machine and even for just an individual user.

Ultimately, the widespread practice of selling computers based upon 64-bit processors but limiting them to 32-bit Windows usage is more than a little dishonest. It is somewhat like a car company selling cars that are advertised with 8-cylinder engines but not making it clear to consumers that the car will be delivered with a fuel injection system and ignition that allows only four of the cylinders to function. Don't let that happen to you. When buying a computer with a 64-bit processor, buy it from a vendor who can deliver it to you with 64-bit Windows. Don't settle for anything less.

**FAQ**

Why do you have separate 32-bit and 64-bit product offerings? Why not simply sell a 64-bit product at one low price? - The nature of 64-bit Windows technology and 64-bit processor technology is such that the creation and maintenance of true 64-bit code requires a significant amount of engineering in addition to what goes into creating the standard 32-bit product. At the present time there are many users of 64-bit Windows, enough to justify the expense of creating a 64-bit product, but there are still many users who continue to operate 32-bit Windows. That means Manifold must be made available for both 32-bit and 64-bit Windows versions. If there were just a single Manifold product that could run either 64-bit or 32-bit then 32-bit users would end up having to pay a higher fee to subsidize the creation of a 64-bit product they do not use. By offering two product versions, a 32-bit and a 64-bit version, we can retain the lowest possible price for 32-bit users while charging a slightly higher fee for 64-bits only to those users who desire the 64-bit version.

Does that mean I have to buy a separate Manifold product to run in 64-bits? Yes. Low cost upgrades to 64-bits are available if you already have a corresponding 32-bit Manifold product.
I'm not ready to jump to 64-bits quite yet but want to buy an x64 Manifold version now. Can I use it in a 32-bit Windows system? Yes. An x64 Manifold edition can also run in a 32-bit Windows system: simply install from the 32-bit installation package and use your x64 Manifold System serial number (which can activate either a 32-bit or a 64-bit installation). When you upgrade your computer to a 64-bit system, install Manifold from the 64-bit installation package and use your x64 Manifold System serial number again.

I have a 64-bit processor but I am running 32-bit Windows XP on it. If I install x64 Manifold will it run using all 64 bits of the processor? No. Manifold can only run in whatever mode the host Windows system allows. If you are running 32-bit Windows you are using your 64-bit processor to emulate a 32-bit processor as far as Windows and any other applications are concerned. Like all applications in that case, Manifold will also run in 32-bit mode. To run Manifold in 64-bit mode on a 64-bit processor you must also be running 64-bit Windows.

Are there real benefits to running x64 Manifold in x64 Windows? Yes, if you have a reasonable amount of memory (2 GB or more recommended). 64-bit Windows is far more reliable than 32-bit Windows, and far faster as well. 64-bit Manifold is visibly faster in most cases than 32-bit Manifold. Don't hesitate about this.

Isn't it a hassle to get device drivers for 64-bit Windows? Not if you are using a well-established 64-bit operating system like Windows XP x64 and you are running reasonably standard hardware purchased from a vendor who is looking out for the interests of their customers. Because of the popularity of 64-bit Windows for gaming, virtually all motherboard vendors catering to gamers (the market driving the best deals for the highest performance) make it a point to package 64-bit Windows drivers by default for their products. Using newer operating systems, like 64-bit versions of Vista, or making the mistake of buying products from vendors who sell 64-bit processors but don't intend to allow customers to use them in 64-bit mode (just about every laptop vendor around) may require some detective work to locate 64-bit drivers.

Isn't Windows Vista always 64-bit? No. Windows Vista is available both in 32-bit and in 64-bit x64 editions. If you want to run 64-bits you must have Windows Vista x64.

Should I use Windows XP x64 or should I use Windows Vista x64? That's up to you, as 64-bit Manifold will run in either XP x64 or Vista x64. If you are planning to use Windows Vista x64 it makes sense to procure it pre-installed on the computer system you are buying, since as of the present writing there is less driver support for Vista x64 than for XP x64. By purchasing Vista x64 pre-installed on a computer you assure that the manufacturer does indeed have full driver support for Vista x64 for their particular configuration of devices. Because XP x64 has been around longer than Vista x64, XP x64 is very widely supported with drivers by many different device vendors. For example, NVIDIA CUDA drivers for XP x64 will quite likely appear before Vista x64 drivers. XP x64 is faster than Vista x64, but Vista x64 is prettier.

If I provide a Manifold x64 product serial number to an existing 32-bit Manifold installation, will it re-launch in 64-bit mode? No. You must uninstall your old Manifold and install the new 64-bit installation package. The 64-bit installation package may be downloaded. If you have not installed a 64-bit installation package the 64-bit serial number provided for x64 Manifold products will not work.

If I provide a Manifold x64 product serial number to a 32-bit Manifold installation, can it continue running in 32-bit mode? Yes, if the serial number is for the correct Manifold release level. For example, you can't provide a 7x 64-bit serial number to a 8.00 installation and expect it to work. Assuming you are using serial numbers that match the release, 64-bit serial numbers will work with either 32-bit or 64-bit Manifold installations.

How can I tell what kind of serial number is currently in use? Launch Manifold and look in the Help - About dialog. That will identify the serial number and the mode of operation for Manifold. If you have shortcuts for both Manifold System (32-bit) and Manifold System (64-bit) then you have installed a 64-bit product. You can always check the status of a particular serial number by visiting the Support page on the manifolds.net web site and visiting the link on that page to the Serial Number Status page.

To switch between 32-bit and 64-bit operation on a 64-bit Windows system, must I re-install Manifold? No. Install the 64-bit version, which then installs two short cuts, one which launches Manifold in 32-bit mode and one that launches Manifold in 64-bit mode. You can go back and forth between 32-bit mode and 64-bit mode simply by choosing which shortcut you use to launch Manifold. This is exactly similar to how Windows x64 editions provide Internet Explorer in both 32-bit and 64-bit versions, either of which may be used at will.

Can I launch two Manifold sessions, one of which is 32-bit and one of which is 64-bit at the same time? Yes, if you have installed a 64-bit Manifold installation package on a 64-bit Windows system.

Are .map project files for 32-bit and 64-bit Manifold editions compatible? Yes. The same .map format is used both for 32-bit and 64-bit Manifold editions.
I have a dual core 64-bit processor. Will x64 Manifold work with that? Yes. If you have a dual core or other multiple core processor Manifold will run fine and those parts of Manifold that are designed to take advantage of multiple processors or multiple cores (such as, for example, the multithreaded rendering architecture used by Manifold to render image libraries) will take advantage of all processors or cores available. For example, if you have a motherboard that has two processor sockets and each socket has a quad core processor in it, Manifold will be able to take advantage of all eight processor cores.

Will Manifold x64 run with Itanium processors? No. manifold.net has no plans to support Itanium. We believe x64 is the way to go, using processors like AMD's x64 processors or Intel's x64 processors.

My runtime application requires access to MDB files. How do I arrange that in a 64-bit Windows system? That can be accomplished by writing your application as a 32-bit application. Installing an x64 edition of Manifold, such as Manifold Professional Runtime x64, on an x64 operating system installs both 32-bit modules and 64-bit modules. 32-bit programs using the Manifold object model will use 32-bit Manifold modules and 64-bit programs will use 64-bit Manifold modules, automatically. Your 32-bit application will use the 32-bit Manifold modules and thus have access to MDB.
Introduction

Manifold is a Geographic Information System (GIS) software package. A GIS package is different than an "atlas" package, such as Microsoft's Streets and Trips or the various DeLorme products that can display only one, built-in, read-only map. A GIS can work with many different maps and can be used to create new maps, to edit maps and to combine maps with database information. See the Key Ideas in GIS topic for how a GIS differs from consumer-oriented atlas products.

Manifold is a "word processor for maps" that will let you create new maps or edit existing maps. It is also a "database system for maps" that will allow you to embed database information into a map so that the map can be used as a visual interface into the data. Anyone can use Manifold interactively to create new maps or to use Manifold to explore database information displayed within a map. In the hands of experienced users with programming skills, Manifold may be used to create new applications to work with maps, images, surfaces and databases.

Manifold includes a huge number of capabilities to work with different types of data at the same time. The summary below sets forth highlights. These are provided as a reality check to help the reader realize the resources he or she has on hand in Manifold, and thus to motivate the reader to invest a day or two into reading this documentation before beginning work. The more of Manifold you learn to use, the better Manifold can serve you.

- **Drawings** - Most digital maps are "vector" drawings like those used to create blueprints in Computer Aided Design (CAD) programs such as AutoCAD. To deal with these, Manifold includes a vast array of CAD-style editing features. These are so extensive that people will use Manifold as a general purpose CAD editor. Specialized features such as tracing ("vectorization") of images provide special CAD capabilities that are required in a mapping environment.

- **Linked Drawings** - Manifold can create linked drawings "on the fly" from data within databases and can show those linked drawings within Manifold projects just like regular drawings. There are many applications where data that has a fundamentally geographic character is stored within centralized databases. An example might be a list of a bank's ATM locations that are maintained in a geocoded database in one of the bank's DBMS servers. It's convenient for the bank to keep that data in some centralized database where other applications, such as archival storage of ATM machine maintenance records, can work with it, but it is also very convenient that Manifold can dynamically create drawings from that data for purposes such as displaying a map of the nearest ATM locations in a Manifold IMS web site. It's also very convenient that any changes made to that centralized ATM database will automatically be incorporated into linked drawings created from that database. Linked drawings can also be used as a visual user interface to edit data by multiple users within centralized databases. For example, a town might maintain tax parcel information in a database and use Manifold via linked drawings as the user interface to that data to display parcels, to allow people to find parcel information, edit the shapes of parcels, add or delete parcels or edit the database information associated with each parcel.

- **Database Management** - Modern maps will link the objects in drawings with records from database management systems. One can then use the map as a visual interface to the data. Manifold includes a full-power database management system complete with SQL and numerous other capabilities to make it easy to work with databases and maps simultaneously. The system includes sophisticated data exploration tools and decision support using variable views, fuzzy logic and Manifold's patent-pending "more like this" technology. Manifold's database facilities are so strong that people will often use Manifold to explore and manage databases having nothing to do with maps. Manifold includes the most powerful and sophisticated spatial SQL engine ever introduced in GIS.

- **Direct Connect to Spatial DBMS** - All Manifold releases can connect to almost any DBMS for attribute storage while Enterprise Edition and above can connect directly to spatial DBMS products like Oracle, Oracle Spatial, SQL Server 2008, IBM DB2 with Spatial Extender and PostgreSQL / PostGIS. No middleware is required to achieve simultaneous access by potentially thousands of users to potentially terabytes of data stored on enterprise class spatial DBMS. In fact, Manifold even provides a spatial extender to convert SQL Server 2005 into a true spatial DBMS and Manifold provides generic spatial DBMS capability that can confer spatial DBMS capability onto virtually any DBMS product, including MySQL and even Access. See the Spatial DBMS topic for an introduction to this amazing capability.

- **Images** - Images in various formats are used throughout mapping. We use images as backgrounds, as semitransparent layers within maps, as sources for creating new digital maps, as graphics arts embellishments for a better presentation and for many other purposes. Manifold includes full capabilities for importing, editing and manipulating images, including alpha transparency capability on a per pixel basis, georegistration for use in geographic contexts and transformation to any geographic projection. The roster of image editing effects is rivaled only by professional graphics editors such as Adobe Photoshop.
Compressed images allow very rapid display of even multi-gigabyte images when used as backgrounds in maps.

- **Linked Images** - Just as with linked drawings, Manifold can create linked images that are created "on the fly" from data stored in external databases or fetched on demand from image servers of various kinds. For example, Manifold can automatically fetch and mosaic images from TerraServer or other types of image servers, such as Google Earth, to automatically provide photographic images that cover a particular region of interest.

- **Surfaces and Terrains** - Manifold can import terrain elevation data from almost any format known and incorporate surfaces into maps as shaded 2D images showing relief, slope, aspect or visualized as realistic 3D terrain views. Terrains may be overlaid with any combination of vector drawings or raster images. Surfaces are also used to provide continuous gradient displays of data such as population, temperature or other variables. Manifold can automatically create surfaces from scattered data points and compute contours, aspect, slope and other features.

- **Raster Data and Images** - Many data sets such as satellite sensor scans of the Earth's surface are published as raster data. These can be used as Manifold "images" for analysis and presentation. There are many names for such data used in GIS, such as "Grid" data, depending on the context of usage. In Manifold, such data sets are either raster images or surfaces. Manifold provides numerous analytic and image processing capabilities to get the most out of raster data.

- **Statistics and Analysis** - Manifold’s solvers and analytic subsystems include hundreds of analytic capabilities in statistics, general purpose mathematics, network analysis, graph theory, database logic, topology and computational geometry. Active Columns™ and ViewBots™ provide a new style of interactive analysis that's easy to use.

- **Charting and Data Mining** - Manifold’s chart windows provide 2D charts for data exploration of information in maps and databases as well as minicharts that appear in drawings and maps for each object. Manifold Charts are a straightforward visual interface for exploring and understanding data. Other data exploration tools include ViewBots™ and the Manifold Decision Support System.

- **Spatial Analytics** - Manifold provides a wealth of spatial analytic capabilities embedded throughout the entire system. Relief effects in 2D images are automatically computed based on terrain elevations from their associated 3D terrains, for example. Manifold SQL includes dozens of spatial extensions, which apply throughout raster images as well as within vector maps. Buffer zones, spatial statistics and numerous other functions provide sophisticated spatial analytics within every system.

- **Programming** - Manifold System includes a full programming environment using Microsoft languages at no additional cost. The power of Manifold System may be extended by writing scripts in Microsoft .NET scripting languages or Microsoft ActiveX scripting languages like Visual Basic Scripting (VBScript) or JavaScript. Free downloads from various sources allow ActiveX scripting in other languages as well, such as PERL and Python. Manifold’s programming environment includes full editing capabilities as well as "drag and drop" creation of programming forms using a wide variety of Windows controls. Manifold also may be programmed from external applications written in Microsoft standard languages such as Visual Basic or Visual C++ to create custom mapping applications based on Manifold. For casual customization, Manifold includes Active Columns and ViewBots for "ad hoc" programming. Manifold script windows automatically color script source code by syntax and support advanced editing "power moves."

- **Debugger** - The Manifold Debugger provides a sophisticated debugging environment. The Debugger allows stepping through multiple scripts with automatic stops at breakpoints, paused execution and stepping into, over and out of routines. The Call Stack, Variables and Watches panes provide dynamic, editable displays of variables, functions, computed values and contexts.

- **Internet Map Server** - All Manifold editions except Personal Edition include a built-in, enterprise-class, powerful Internet map server, Manifold IMS, which can publish your maps to Internet for people worldwide to view. The map server allows browsing, panning, and zooming within maps that you choose to publish as well as support for queries, geocoding, hyperlinks, information tools and layer selection if desired. Advanced users can customize the map server to create spectacular Internet pages. Manifold IMS is easy to use with no programming required. Use IMS to create web pages for your organization's internal network so that other people within your organization can see or themselves the value of what you do in GIS without the need to drag them one by one to your computer to see a demo. Manifold IMS works with any Windows HTTP server and provides pre-built templates for use in ASP and ASP.NET Windows IIS environments. Manifold IMS provides not only a classic HTTP style internet map server but also an OGC WMS server, an OGC WFS-T (WFS with transactions) server, and a Manifold Image Server for a complete range of GIS-enabled web technologies.
Sophisticated Image Linking and Serving - Manifold can create linked images from ECWP servers to provide high-speed, multi-resolution imaging, and Manifold can link images from OpenGIS Consortium (OGC) WMS servers, operating as a client. In addition to operating as a client, Manifold IMS can also function as an OGC WMS server, generating on-the-fly WMS images to be served to other clients that can operate using OGC WMS protocols.

Image Libraries - Manifold can automatically tile together collections of images from a wide variety of formats to form a single image mosaic. Image libraries make it easy to utilize collections of images covering broad regions as a single background image. Automated creating of index drawings provides the infrastructure for easy navigation and utilization of image libraries in applications such as IMS web sites.

Dynamic Geometry using Industry Standards - Manifold can read and write object geometry in connections to external database systems to create linked drawings from tables or queries. Manifold can use ESRI-style "geodatabase" geospatial geometry, OGC-style "well-known binary" WKB geometry as occurs within Oracle Spatial and other DBMS products Manifold's own high performance Geometry data type that can be stored in any leading DBMS product such as SQL Server, Oracle, IBM DB2 or MySQL. Manifold Enterprise Edition and above also supports the Oracle Call Interface for geometry storage using Oracle native technology in Oracle Spatial and Locator technologies as well as image storage using Oracle's GeoRaster technology.

Spatial SQL for Images and Surfaces - Manifold's Raster Extensions to spatial SQL enables use of "virtual tables" to create, analyze, modify and display raster data within images and surfaces. Images and surfaces can be linked from data stored in external DBMS providers using almost any leading DBMS product.

Import and Export - Manifold System can read data from over 80 different vector, raster, terrain elevation and database formats including specialty formats such as grids and the very latest government formats such as HDF EOS used with ASTER Earth observation satellite data. This allows users to access terabytes of free GIS data by Internet directly from the source, and to use data from every GIS system known as well as from numerous specialty software packages. The Professional Edition writes to standard interchange formats while the Enterprise and Universal Editions includes additional export capability for specialty formats of interest to larger organizations.

Enterprise Edition - If you have licensed Enterprise Edition, your edition of Manifold will also provide centralized storage of components within databases such as Oracle, IBM DB2 or Microsoft SQL Server or SQL Server Desktop Engine. In addition, Enterprise Edition provides a coordinated framework within which many users can share the same components, allowing editing of shared components without fear of simultaneous, incompatible edits by different users. Enterprise Edition also enables direct use of Oracle technologies such as Oracle Spatial, Locator and GeoRaster technologies.

Optional Business Tools - If the optional Business Tools extension has been installed, a variety of new commands will be added. Business tools provide advanced districts and visual districting commands, drive-time zone analysis, optimal routes and the ability to send email directly from Manifold for geo-targeted emailing.

Optional Geocoding Tools - When the optional Manifold Geocoding Tools product is installed, the geocoding engine built into Manifold System can find the geographic location for street addresses and ZIP codes in the United States. Use the geocoder to automatically plot the locations of addresses, to find distances and to execute spatial queries using the geocoding functions in Manifold SQL. The Manifold geocoding engine can be used from within IMS applications to address-enable web sites created with Manifold IMS. The Geocoding Tools package also adds the ability for fast and easy import of drawings from the Geocoding Database provided by download.

Optional Surface Tools - If the optional Surface Tools extension has been installed, Manifold will gain the ability to work with new Profiles and Elevations components as well as new commands and other capabilities. These include a sophisticated surface transform dialog, a transfer heights command and greatly extended surface interpolation capabilities for triangulation, Kriging, and Median-Polish Kriging. Surface Tools also adds sophisticated watersheds capability to enable computation of water sheds, streams, flow accumulation and other hydrological features.

Universal Edition - If you have licensed Universal Edition you have all Enterprise Edition features as well as all Business Tools, Geocoding Tools and Surface Tools features.

Database Administrator Edition - Manifold Database Administrator Edition is a version of Manifold System that includes additional capabilities for setting up and managing GIS data storage in DBMS systems such as Oracle Spatial. Database Administrator Edition includes tools such as Batch Export and Administrator Console that allow DBMS storage to be configured with capabilities like friendly names, support for
formatting and pre-configured import and link options that will thereafter make life easier for other Manifold users.

- **License Server Edition** - The Manifold System License Server product gives organizations the ability to maintain an inventory of floating Manifold licenses that may be used by any client computer that can connect to the License Server, all without needing to use serial numbers and Activation keys for individual client licenses. License Server provides the economy of floating licenses as well as greatly reduced administrative overhead for organizations that deploy many hundreds or thousands of Manifold licenses.

**Key Ideas**

Although the functional areas covered by Manifold capabilities are very broad, they have been implemented in Manifold System using a common set of user interfaces and a common set of key ideas, such as selection. The key to mastering Manifold is to understand those key ideas.

If you are new to GIS or wish a refresher in GIS from a manifold.net perspective, you must first get acquainted with a few basic GIS concepts. Please begin by reading the Key Ideas in GIS and Key Ideas topics in the Appendix.

After reading all of the topics in this Introduction section, please read the more detailed sections that are organized by component or activity. A general reference to nearly all menus, dialogs, panes, toolbars and context menus occurs in the Menus, Dialogs and Controls Reference section. The best way to begin working with Manifold System is to follow along with the many example topics in the Examples chapter.

**A Note on Illustrations**

Some illustrations in this manual use screen shots based on "classic" Windows appearance such as used in Windows 2000 and some use a more modern appearance such as used in Windows XP. When installed in Windows XP Manifold will automatically configure the user interface to use Windows XP style for controls, as seen below in a screen shot of the column formatting dialog for table windows.

![Column Format Dialog](image)

Depending on your Windows settings for Appearance choices in the Windows Display Properties dialogs, your desktop may be configured to use different colors, fonts or styles from those seen in screen shots used in this manual. Color and stylistic choices made for illustrations are not necessarily Windows defaults but have been chosen for maximum clarity and legibility.
Key Ideas in GIS

Read this topic if you have never worked with a GIS before or if you would like to see the manifold.net spin on basic concepts in GIS. If you already know GIS well, skip ahead to the main Key Ideas topic.

This topic introduces foundation concepts that apply to any modern GIS and which must be understood to use Manifold. We will explain these concepts by making analogies to software with which the reader is already familiar. For example, we will use Microsoft Word to make analogies to word processing software and AutoCAD to make analogies to CAD programs.

This is a very long topic, possibly the longest, wordiest and least-illustrated topic in this Help system. Don't panic. Almost all of the other topics you will encounter will be easier to read than this one.

Key Idea: Manifold is a GIS.

GIS is an acronym for Geographic Information System, which is software industry jargon for a program that combines a variety of mapping and editing functions together with database capability. Many of the functions of a GIS appear in other sorts of programs, such as CAD editors like AutoCAD, or database programs like Microsoft Access. It is the unique combination of capabilities within a GIS that gives this class of applications their unique power.

As a class of software applications, GIS has been around for about thirty years. Only recently have GIS applications appeared on desktop machines. As a result, only recently has the traditional GIS industry begun to undertake the social, technical and economic transition from minicomputer and mainframe cultures to Wintel clone culture that has occurred throughout the rest of the software industry.

Manifold is the first "Wintel" GIS ever created. It was deliberately created to focus exclusively on Wintel architectures, to leverage mass-market standards like Microsoft scripting languages and Windows interfaces, and to attack sleepy traditional markets with a mix of very high performance and very low cost.

Pronunciation: Most people in English pronounce the acronym GIS by saying its letters: "Gee Eye Ess." In other languages, the acronym is often pronounced as "Jis" using a soft "J" sound like in "Jester" or "Gis" using a hard "G" sound like in "Gate".

Key Idea: A GIS is like a CAD editor or word processor for maps.

We can use AutoCAD to create elaborate blueprints and diagrams. We can use Microsoft Word to create elaborate documents. In both cases, we can draw new items, we can edit existing items, and we can import items already created from other documents. With AutoCAD we can enjoy a host of editing capabilities that allow us to draw diagrams and blueprints more easily. For example, we can set "snap" characteristics so that the mouse cursor will automatically jump to the ends of existing lines. This makes it easier to draw lines that begin and end exactly at the ends of existing lines.

A GIS allows us to:

- Import existing maps. A good GIS will let us import existing maps from many different file formats, including those used by competitive software.
- Create new maps, and edit existing maps. This includes adding new items to a blank map, or to change the shape and position of items already in the map or to delete items from a map. For example, we might edit a map of counties in a particular state to show only three counties if that is our region of interest.
- Change the appearance of items in a map by changing their formatting.

The above capabilities are similar to any CAD or graphics editor, or, for that matter, any good word processor. In addition, a GIS has other capabilities. Using a GIS we can:

- Specify the geographic location of the map and the items in it.
- Use geographic projections to show a more realistic view of the map.
- Link the items in the map to a database table. This allows the graphics in the map to act like visual "handles" to reach into the database.
• Process the data and visual items in a map to create new items or to analyze the data. For example, we might create a "buffer zone" that shows all regions within 100 meters of a road or stream, or we might automatically add up populations of a given species of tree within a set of tree stands in a forestry application.

The above capabilities set GIS programs apart from ordinary CAD editors like AutoCAD or graphics editors like Adobe PhotoShop or Illustrator. Anyone can use PhotoShop to draw a map that is a fine illustration. However, that illustration is just a picture. There is no "intelligence" in the picture that connects a particular spot in the picture to a real geographic location. We cannot, for example, draw a picture of Europe in PhotoShop and then use that picture to find the precise latitude and longitude location of Geneva, Switzerland.

We could use a CAD editor to create a beautifully detailed diagram of streets in San Francisco. However, we could not automatically have each street linked to a record in a database table that for each street would have fields storing the name of the street, when it was last paved, what paving material was used and so on. All we have is a pretty and highly detailed diagram, but no linkage to a database.

With a GIS, on the other hand, we could create that diagram of streets, because a good GIS will have the same CAD drawing capabilities as any good CAD editor. In addition, we could place any drawing we create within a true geographic location and we could fill in the fields for each street line's record in the database table. We could then use this information in spectacular ways.

For example, we could find all streets within a mile of a given location that have been repaved in the last three months and then we could sort the database records for those streets by which street was most recently repaved. We could use SQL to find all the streets in the map that were last paved with concrete and highlight them in the drawing. We could show points on the drawing that are located at latitude and longitude coordinates taken from a GPS device that logged the movement of a bus. We could automatically calculate our costs for different time periods when paving will be necessary in the future. Once the system has a handle on both visual appearance and shape, location and database information the possibilities are virtually endless.

A GIS is therefore most different from static, purely visual applications like AutoCAD or Photoshop in that it is both a visual representation as well as a data representation, and it is at once both a presentation and visualization tool as well as being a control console and an analytic device, a limitless calculator of appearance, location, shape and data content.

**Key Idea: A GIS can work with many different maps.**

A GIS can work with many different maps. Just as we can use Microsoft Word to open any document in a file format understood by Word (such as Word Perfect or other word processors), we can use our GIS to open maps saved in any file format understood by the GIS. We can create our own maps and we can save them for later use or we can use maps created by others.

**Nomenclature:** When we use the word "map" in this topic we mean a digital document that is a drawing in some GIS format that may or may not also include database information. The phrase "GIS format" means some file format that is useful for saving maps in a way that's useful to a GIS. A real GIS, for example, will save its files using a file format that allows saving database information, projection information and other necessary details. Although people often will use a CAD program like AutoCAD to draw diagrams intended for use in maps, AutoCAD's .dxf format is not really a GIS format since it is clueless about geographic location, projections or database information. One can often import .dxf drawings into a real GIS and then manually add projection and database information using the capabilities of one's GIS but that is obviously not as convenient as having everything done up front.

We are often very interested in having everything done for us up front with maps in a nice, neat GIS format because sophisticated maps are often highly detailed drawings of immense complexity. Creating such a map from the ground up is not a job for the faint-hearted. Consider drawing a map that shows all the water features (streams, lakes, rivers, etc.) in, say, Texas. Creating such a diagram in AutoCAD or Adobe Illustrator would be a massive undertaking, requiring the creation of hundreds of thousands of lines in a precise way. Using a GIS can make the process easier, but it is still such a massive and expensive project that quite likely only the government can afford to do it.

So, even tough a GIS can be used to create new maps line by line we will probably spend our time searching Internet to find a pre-built map we can use. We can then use the GIS to change that map, to format it into pretty colors we like, add new features of special interest, combine it with other maps, or cut and paste features from the map into other maps. Editing and cutting and pasting is a lot easier than creating something from the ground up.

Because maps used with GIS can be so complex and such a hassle to create from the ground up, sometimes our ability to accomplish a specific project will depend on whether or not we can find the right pre-built maps or data.
sets to use. For example, suppose we have a database that lists the zip (postal) codes of customers who have called our 800 number to place an order for a product. We might want to display a map of the United States showing zip code regions with each region colored in to indicate how many customers called from that zip code last week. Zip codes with high numbers of calls might be colored bright green and those with lower numbers of calls could be colored in progressively darker shades of green until those zip codes with zero calls would be colored black. Right away, we could see on the map the geographic locations of high call activity.

That's a great idea and easy to do in Manifold (or any other modern GIS) if we have a map of zip codes as regions in the United States. If we can locate such a map we can do the project in about five minutes. If we don't have such a map, we can't do the project. It turns out that getting a map of zip code regions in the United States is not so hard to do, since we can download one for free from the US Census Bureau's web site. However, such maps of postal code regions for countries other than the US are extremely difficult to download for free from Internet. If we could not locate such a map for free on the Internet we might have to buy it from one of the commercial vendors who are in the business of supplying specialty maps and demographic data for GIS users.

Manifold's contribution to lowering the cost of acquiring maps is to provide import capability for essentially all GIS formats in use. If you can find it on the Internet for free in almost any form, it's almost guaranteed that Manifold can import it. In particular, Manifold can read all of the exotic federal government formats in which many maps are published for free. Because there are millions of maps and other GIS data sets that can be downloaded for free from Internet (many of which may be downloaded from government web sites for which your tax money has paid), this gives Manifold users access to the largest library of GIS information of any GIS system in history.

Key Idea: Manifold is a GIS, not an atlas.

Even if you have never used a GIS before, it is quite likely you have used some sort of mapping program such as Microsoft's MapPoint, Streets and Trips, or any one of a series of DeLorme products. Although such products display maps, they are best understood as atlas programs and not as GIS programs.

Atlas programs are mapping programs that work with one, built-in map. Microsoft Streets and Trips, for example, is a program based on the display of a built-in map of streets in the United States. It's easy to use because it is designed to work with that one, pre-formatted map. However, we cannot use Streets and Trips to create new maps nor can we use Streets and Trips to look at, say, a geologic map of the United States that we may have found on the Internet or at, say, a map of ocean depths off the coast of Corsica.

In contrast, a GIS like Manifold can be used to view many different maps from many different sources. We can use Manifold to create entirely new maps or to edit and combine existing ones. This is one of the major differences between a true GIS program and an atlas program.

This difference between a true GIS program and an atlas program is similar to the difference between Microsoft Word and an Adobe PDF document viewed with Adobe Reader. The Adobe PDF file delivers a "canned" document to a user. We can view the PDF file but we cannot change it. The PDF file is easy to use because it allows no changes - someone else did all the work of creating the text and formatting it in a pretty way. Double click on the PDF file and Adobe Reader will display it as intended. But that PDF file is forever limited to the one document it contains.

In contrast, Microsoft Word can open documents in any format understood by Word. We can create new text and new documents in Word, and we can use the extensive array of formatting and stylistic features in Word to give our documents exactly the appearance desired. We can even add images to our documents. Because Word can do a lot, learning Microsoft Word requires more thought than learning how to click open an Adobe PDF file. The additional thought required to learn and operate Word is rewarded with the freedom to create whatever document we like.

In the world of mapping software, atlas programs like Streets and Trips are like a PDF file; they provide one, built-in map that has been cleverly formatted by the authors of the program but which cannot be changed. An atlas map program automatically starts up with a specific map full of street symbols and roads and lakes and streams. That's certainly very handy if the sort of map you wish to use is exactly the map that is used within the atlas program; however, if you would like a different sort of map an atlas program cannot help you.

A common complaint from users of atlas programs like MapPoint or DeLorme products is that they would like to create maps that are different from the built-in map. For example, users might want to create a presentation that shows only one or two states that are the subject of a particular presentation, and they would like to remove all other states to provide a cleaner presentation. For that, a real GIS is necessary.
The illustration above shows many states, which can be confusing if we only want to show a presentation involving four states.

It takes mere seconds in a GIS package to edit it so that only the four states of interest appear but that is not possible in atlas programs. We can even add labels or have them automatically generated from database information, or otherwise edit the display as we see fit:

The GIS program’s ability to manipulate visual data as desired, to add items like labels in a full-custom way and to otherwise edit the display makes it a lot easier to create visual displays that are easy to understand and which have great impact. Even in simple things like the above illustrations, a GIS program can do what atlas programs cannot, and that is even without considering the immense power of linking database information to a visual interface.

A GIS program like Manifold is for maps and data what Microsoft Word is for text documents: it is a “word processor for maps.” Manifold can open nearly any type of document containing map data and it can be used to create almost any sort of new map. It can edit existing maps, combine them, cut and paste between them and otherwise perform a vast number of editing, formatting and analytic functions.

Key Idea: Mapping programs and map data are different things.

Atlas programs like Streets and Trips or MapPoint feature very tight integration between the functions of the viewing program and the map that it displays. There will even be “hardwired” menu choices in the program’s menu that refer to specific items in the map. For example, one might see a choice of menu items to “Show Streets” or “Show Hotels” that turn different items in the map on or off for viewing.

This tight integration is handy when one wishes to work only with one map, but it often leads newbies into confusing the program with the data it shows. Consumers will refer to Streets and Trips, for example, in a way that does not distinguish the content of the map data from the capabilities of the display and user interface program that shows us the data. They’ll say things like “When I click open the Streets and Trips map, the map won’t let me draw on the map.”
In GIS, both newcomers and even GIS people will at times confuse the GIS with the data upon which it works. Newcomers tend to make this confusion because of their previous familiarity with atlas-style, consumer map programs as discussed in the preceding section. Some GIS "old hands" will make this mistake because they are still stuck in the GIS technology of an earlier era.

In earlier times, GIS programs were so expensive and so difficult to use that very few people ever worked with a sophisticated map/database editor. Instead, most people who encountered a GIS or used mapping data would see the data already integrated with the GIS software. People would say things like "I want to create a GIS of Camden County" and by that would mean that they wished to create an integrated map, database, and viewing method. In earlier times, "GIS" did not mean just the software: it meant an ensemble of both software and data. In modern times, "GIS" now means the mapping / database program that is used to view, edit and manipulate maps and data of all kinds.

To the new user, this key idea may seem to be a small distinction without much meaning. However, when new users import maps into a GIS and commence using those maps, they may encounter all sorts of unexpected effects caused by the structure of those maps.

It's important to keep clear in one's mind the difference between the GIS program's function and any effects caused by the particular data set with which one works. For example, if one is working with a drawing that is made of lines only and has no area objects in it, then one cannot color the regions between the lines. That's not a limitation of the GIS. It is simply a reflection of the structure of the drawing that one has chosen to use. We could choose to use a different drawing, or we could use Manifold to create area objects if we wanted to color them.

**Key Idea:** Formatting changes the appearance of objects in maps.

Digital vector maps, the most common type of GIS map, are like CAD drawings in that they are "vector" data sets. The items in a drawing, such as a shape that shows an island or a line that shows a road are called objects. This word is not intended to connote any fancy meaning, such as the term "object" does in programming... it is simply a word that means "thing" or "item" and is used to mean the points, lines or areas that make up the content of the vector drawing, excluding, however, annotations such as labels or accessory elements such as legends, North arrows, scale bars, graticules of latitude / longitude lines and so on.

All objects seen in the map are drawn using points, lines or areas between precise coordinates in a connect-the-dots fashion. By default, such objects will appear in whatever default colors and display styles are used by the GIS system. To alter their appearance to make the map more legible, informative or prettier we can change the formatting of the objects. We can also add accessory elements such as labels.

Formatting in a GIS context is directly analogous to formatting in a word processor such as Microsoft Word. When we format text in a word processor we apply display attributes such as font style, font size and color to the text.

The quick brown fox

**The quick Brown fox**

If we import text into a Microsoft Word.doc it might first appear in Courier New font. We can format the text to change the appearance. Changing the font size, style and color can result in dramatic changes in the appearance of the text but the actual words stay the same. Just so, changing the formatting of objects in drawings can change dramatically their appearance even though the actual objects within the drawing stay the same.

In Manifold System, we will often import complex maps from formats that do not retain information about visual appearance. In such cases, Manifold will initially show the drawing using default formatting. Specifying formatting to create a pretty map from such raw drawings will require some work and good taste.
Both of the illustrations above show exactly the same drawing, which contains vector objects (areas, lines and points) that represent land areas, depth contours in water, road line, very small areas showing certain populated regions and points marking the locations of named populated places. The illustration on left uses default formatting. The illustration at right has had formatting applied so that land areas are colored green, ocean areas and depth contour lines are colored blue, road lines are shown in black and population points are shown as round yellow dots with the tiny populated regions in small areas of yellow color. Note how the application of sensible formatting makes the image on the right more comprehensible.

Formatting is often confusing to beginners who are familiar only with consumer atlas programs. They don’t realize that one reason the atlas program displays a pretty map is because a team of people spent many hours choosing good formats for the data set being used. In fact, since the atlas program is dedicated to displaying only one map, it may have special display features in it that facilitate clever formatting. The classic example is a special formatting engine that draws roads as thick lines with contrasting color borders, and which correctly splices the border edges at road intersections.

It's reasonable to wonder why a GIS program might not have a "built in" format that automatically makes maps very pretty. After all, if a $20 consumer atlas program can show a pretty map, why does a fabulous, state-of-the-art GIS program show maps by default in a stick-figure, simple style?

The reason is that because a GIS can open any map there is no way for the program to know how we would like that particular map to look until we tell it. One day we might work with a map of lines that are meant to be streets and another day we might load a map of lines that represent geologic faults along the Mid-Atlantic Ridge. It is very easy to create a fixed format for a fixed data set of the sort used in Streets and Trips. It is very difficult to create a format engine that can read our minds and make whatever map we load look pretty in a "correct" way.

Of course, once we format a map we can save it so that it will retain its appearance when reopened.

**A note on nomenclature:** As in most software, the word **format** has two meanings in GIS. One meaning is the scheme of organization of a computer file. One might say "Is that database in Access .mdb format or is it in dBase .dbf format?" Another meaning is the application of effects to change the appearance of something. In word processing we might say "I have change the font format so that it is all boldfaced Arial font in 8 points." In GIS we might say "All the points in that layer are formatted so they are small blue stars." It is usually obvious from the context of use which of the two meanings is intended.

**Key Idea:** A GIS links maps with databases.

In their visual appearance and diagram / map editing capabilities, GIS's are similar to a CAD editor such as AutoCAD. AutoCAD can also be used to create and edit maps. A GIS like Manifold adds another important capability: it incorporates database capability within the program. That database capability is used to link objects in the map with one or more database tables. The GIS links each object in the map to a record in the table.

This is probably the most powerful idea in GIS, the idea that a map can have the objects it shows linked to a database table. This linkage makes it possible to use the map as a visual interface to fetch and manipulate data in the database. In many important GIS applications the visual appearance of the map is a secondary consideration. What is most important is the ability to use the map as a visual interface to deal with the data "behind" the map.

The linkage also enables us to use the map as a presentation tool to make sense of otherwise incomprehensible data. For example, we might click open a map's table and sort the table by a column such as "population" or "sales" or some other field of interest. We could then select the top ten records in the table and see them highlighted in the map. Right away, we can use the map to display data that otherwise might be buried within endless database records.
Beginners often wonder how the data "gets into" the table and how the links are maintained with objects in the map, and whether a database system like Access must be installed on the machine. For now, leave that up to Manifold [the main documentation topics cover all details]. Manifold includes full database capability. No other software is required.

**Key Idea: Importing maps at times requires extra thought.**

This idea may sound discouraging, but it is actually a reflection of the immense choice and variety one gets with a real GIS.

Earlier we introduced the idea that Manifold can import maps and data from many different file formats. We noted that this was a good thing because it is a lot easier to grab cool maps from Internet for free than it is to create new maps. The good news is that Manifold can almost always import data in a usable way. The bad news is that some data is published using file formats that are so old or so inappropriate for conveying GIS data that they require extra work on our part.

Just like Microsoft Word can read many different document formats, Manifold System can read mapping data in many different formats. The analogy to word processing teaches much that is relevant in GIS. For example, although it is usually simple to import text documents from different formats into Word not all word processing document formats may be imported effortlessly.

For example, many people have experienced frustration when trying to import UNIX or Linux documents into Word (with weird solid block characters where the ends of paragraphs should be…) and have seen extra spacing between lines appear at the end of paragraphs when importing from or exporting to documents in HTML formats. Imported documents might use fonts that we do not have on our system so substitutions must be made. Experienced Word users will learn the idiosyncrasies and limitations of different formats they encounter on a regular basis so they can make adjustments when importing documents from those formats.

It’s exactly the same situation with map formats. Maps and mapping data are published in a very wide array of formats, some of which are very old and full of strange limitations and legacy idiosyncrasies. When importing map data from different formats we need to be aware of the limitations of those formats. Mapping data tends to be more complex than word processing data so there are many opportunities for the limitations of old formats to require extra work when importing from those formats.

In the case of a few modern formats, we can usually quite effortlessly import the core data for the map. However, even with modern formats it is very rare that we can duplicate the formatting and stylistic appearance (line style and color, for example) from one mapping program to the next. In GIS work, therefore, the main focus is to import the raw data of a map and then to apply whatever appearance is desired after the map is imported.

One annoying aspect of importing maps is that much map data has been published as projected maps using very old, technically obsolete formats. Some legacy map data formats do not save projection parameters. When working with projected maps in such formats, the necessary information therefore must be added manually based on notes provided in a "readme" file or other accompanying documentation.

Manifold has a very wide range of converters and options to deal with such legacy idiocy; however, some old formats used by legacy systems are so stupid that there is no avoiding the need for extra thought when dealing with such things. In particular, when working with older formats one must keep an eye out for any accompanying documentation that will be required later to make sense of the data.

Another way in which importing maps may require thought is that sometimes we must choose wisely which maps we will use in a given project.

There are often very many different maps of the same subject matter that have been prepared by different agencies. So, for example, there are often hundreds (if not thousands) of digital maps of the United States that may be downloaded via the Internet to show the same subject in different ways. If one is interested in a map of US shorelines there are many, many different maps that show the shorelines at various levels of detail with varying accuracy. Such maps have been created by different agencies at different times using different data sets for different purposes. So, although they all will correctly show Long Island and other parts of the US more-or-less correctly the maps may differ in various small details, their level of detail and so on.

Part of the total freedom of using a real GIS like Manifold is the responsibility of deciding which data sources to use for which projects. If all we want to do is illustrate some nationwide demographic trend on a map of US states it rarely matters which map we choose. On the other hand, if we are combining data from county maps and state maps from various sources to create a detailed wetlands map of Chesapeake Bay we likely will want to be more careful in the choices of maps we use.
For example, we might wish to use both state maps and local, high-resolution maps downloaded from the USGS servers. This would make it more likely that the shorelines would line up in a neat way with the state boundaries since the same agency prepared them. In contrast, if we used state boundary maps taken from the Bureau of Transportation Statistics (which has a greater interest in roads than in physical geography) and combined them with hydrology maps from the USGS, it is possible there will be subtle incompatibilities such as small misalignments at the boundaries between states or other features in the map.

This can be slightly confusing at first to a beginner, but GIS users quickly discover favorite Internet sites from which to download data and rapidly form strong opinions about which data they like and which data sets they don't like. It's not unusual for someone who has only used Manifold for a week to feel so much like a veteran that he or she will post answers on Internet GIS forums to beginner questions about good download sites for data.

**Key Idea: A street address is not a location**

The only way to locate something in a GIS system is to place it in terms of a specific latitude and longitude location. Street addresses are not the same as a specific latitude and longitude location. For street addresses to be usable within a GIS, they must first be **geocoded** so that each street address record also has latitude and longitude values showing where that address is located. For example, the following table is not geocoded, because there are no latitude and longitude values for each record.

<table>
<thead>
<tr>
<th>Name</th>
<th>Address</th>
<th>City</th>
<th>State</th>
<th>Zip</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sushi Ya</td>
<td>300 University Ave</td>
<td>Palo Alto</td>
<td>CA</td>
<td>94301</td>
</tr>
<tr>
<td>Miyaki</td>
<td>140 University Ave</td>
<td>Palo Alto</td>
<td>CA</td>
<td>94301</td>
</tr>
<tr>
<td>Higashi West</td>
<td>632 Emerson St</td>
<td>Palo Alto</td>
<td>CA</td>
<td>94301</td>
</tr>
<tr>
<td>Naomi Sushi</td>
<td>1338 El Camino Real</td>
<td>Menlo Park</td>
<td>CA</td>
<td>94025</td>
</tr>
<tr>
<td>Tengu Sushi</td>
<td>700 Welch Rd</td>
<td>Palo Alto</td>
<td>CA</td>
<td>94304</td>
</tr>
<tr>
<td>Minokichi</td>
<td>150 University Ave</td>
<td>Palo Alto</td>
<td>CA</td>
<td>94301</td>
</tr>
<tr>
<td>Le Poisson Japonais</td>
<td>642 Ramona St</td>
<td>Palo Alto</td>
<td>CA</td>
<td>94301</td>
</tr>
<tr>
<td>Jidai-Ya</td>
<td>330 Lytton St</td>
<td>Palo Alto</td>
<td>CA</td>
<td>94301</td>
</tr>
<tr>
<td>Akasaka</td>
<td>925 El Camino Real</td>
<td>Menlo Park</td>
<td>CA</td>
<td>94025</td>
</tr>
<tr>
<td>Mikasa</td>
<td>1010 El Camino Real</td>
<td>Menlo Park</td>
<td>CA</td>
<td>94025</td>
</tr>
<tr>
<td>Toshi's Sushiya</td>
<td>211 El Camino Real</td>
<td>Menlo Park</td>
<td>CA</td>
<td>94025</td>
</tr>
</tbody>
</table>

It is easy to make the conceptual mistake of thinking of a street location as being an exactly defined location, the same as a latitude/longitude location. However, that mistake arises mainly from how people use addresses to find locations for the delivery of mail or to go to a particular restaurant or other location.

Street addresses, of course, do not really convey an exact latitude and longitude for the address. They simply provide a means by which a postal carrier or someone else physically traversing the streets can find a particular address. To find an address we have to find the street (with the help of a map if we don't know a particular town), orient ourselves to the address system used on that street, and then locate the address. As anyone who has tried to find an out-of-sequence address in an unfamiliar town knows, there is a great difference between hunting down a particular street address and going directly to a latitude/longitude location.
It is one thing to be able to find a given street address by physically going there (perhaps with the help of a local street map) and it is quite another thing to plot a table of street addresses, such as a table of restaurant addresses, on a map as seen above without ever going to the actual address. To plot each restaurant shown in the table we need to know the actual latitude and longitude address at which it is located. To do that, the table must be geocoded as seen below.

<table>
<thead>
<tr>
<th>Latitude</th>
<th>Long..., Name, Address</th>
<th>City</th>
</tr>
</thead>
<tbody>
<tr>
<td>37.4469</td>
<td>-122.1503 Sushi Ya</td>
<td>Palo Alto</td>
</tr>
<tr>
<td>37.4439</td>
<td>-122.1537 Miyaki</td>
<td>Palo Alto</td>
</tr>
<tr>
<td>37.4435</td>
<td>-122.1514 Hgushi West</td>
<td>Palo Alto</td>
</tr>
<tr>
<td>37.4555</td>
<td>-122.1864 Naomi Sushi</td>
<td>Menlo Park</td>
</tr>
<tr>
<td>37.4364</td>
<td>-122.1730 Tung Sushi</td>
<td>Palo Alto</td>
</tr>
<tr>
<td>37.4439</td>
<td>-122.1536 Minokichi</td>
<td>Palo Alto</td>
</tr>
<tr>
<td>37.4441</td>
<td>-122.1607 Le Poisson Japonais</td>
<td>Palo Alto</td>
</tr>
<tr>
<td>37.4470</td>
<td>-122.1627 Jula-Ya</td>
<td>Palo Alto</td>
</tr>
<tr>
<td>37.4523</td>
<td>-122.1810 Akasaka</td>
<td>Menlo Park</td>
</tr>
<tr>
<td>37.4530</td>
<td>-122.1814 Mlase</td>
<td>Menlo Park</td>
</tr>
<tr>
<td>37.4465</td>
<td>-122.1743 Toshi's Sushiya</td>
<td>Menlo Park</td>
</tr>
</tbody>
</table>

In recent years the adoption of geocoding technology by consumer computer applications has also encouraged us to think of street addresses as being equivalent to a latitude/longitude location for the purpose of computer mapping. Internet mapping sites allow us to enter a street address, such as "525 Main Street, Central City," and instantly see a street map with the location of the address marked as if we had provided an exact latitude and longitude location. Low cost navigation systems that combine GPS technology with built in maps and street address geocoding systems allow us to specify a street address and navigate directly to that location, again, as if we had given exact latitude/longitude coordinates for our desired destination.

As a result, it is quite common for people to expect to be able to enter a street address into a web site or a map and to see a physical location for that address, a sort of "geocoding on the fly." Some applications may give the appearance of taking a list of addresses and displaying them straightaway as points in a map. However, in all cases the software will internally take the intermediate step of using the address to determine a latitude and longitude location for the record. The latitude and longitude location is then used to plot the location of the point.

Software packages use many different strategies to geocode street addresses into latitude and longitude locations. The basic approach is to maintain a large database of streets and address ranges so that the location of a particular address can be estimated from the database. Software that can perform street address geocoding may be built into a GIS package, it may be sold as separate geocoding software, or it may be provided as an Internet web service.

Manifold includes street address geocoding capability for the United States as a built-in capability of Manifold System. The Manifold street address geocoding engine becomes functional when the Manifold US streets geocoding database is installed. If we have installed the Manifold US streets geocoding database on our system we can take a table that contains valid US street addresses and geocode the table to the approximate positions of the address.

However, when using any street address geocoder it is important to understand that the output of the geocoder is an approximate location. See the About Geocoding topic for why this is so.

**Key Idea:** Images are poor substitutes for drawings in GIS.

So far we've used the word map as if it always means a type of drawing or diagram that showed geographic things. It's time to expand our awareness by learning there are really two types of data in common use in the mapping world: images and drawings. The main Key Ideas topic discusses the difference between images and drawings in an overview way and the various topics in this documentation discuss images and drawings in ever-greater detail and sophistication as we read further through this Help file.

Images are the familiar .jpg, .bmp, .tif and similar images we encounter in ordinary graphics arts work. When we scan a photograph we create an image, and when we draw something in Microsoft Paint or Adobe PhotoShop we create an image. Images are also known as rasters, a very old word arising from the television industry back in the dawn of time. Images are made up of pixels arranged in neat rows.

29
Drawings are created by CAD editors like AutoCAD and by cool GIS programs like Manifold. They are made up of points, lines and areas defined by precise coordinate points in a connect-the-dots fashion. Drawings are also known as vector data sets, the word "vector" also being very old jargon taken from the days when primitive computer monitors worked like radar monitors or oscilloscopes.

Drawings have an immense advantage over images in that the items that appear in drawings are readily identifiable (to the computer) "objects" that can be easily manipulated, moved, copied, pasted, edited and linked to database records. In contrast, there are no objects in images, only a sea of pixels. However, images in the GIS and mapping business have one seductive aspect that lures people into using them. Images are easy to make by scanning paper maps in a cheap PC scanner. In a world where most maps are still paper maps, there is a lot of appeal to the idea that one might be able to create a GIS map by simply scanning in a paper map.

Unfortunately, the idea is mostly false appeal. As anyone who has operated a scanner knows, scanning a photograph or map into a PC results in huge files (easily over a hundred megabytes) that have very limited utility. True, the resultant images can be displayed easily enough but they are virtually impossible to manipulate. Try, for example, scanning a road map into a PC and then using a graphics editor to grab a particular set of lines (like highways, or terrain contours, or the borders of a national park) and paste it into some other document. Even with a highly sophisticated graphics editor such as PhotoShop this is almost impossible to do.

The reason why is that we would be attempting to manipulate an illusion. There are no objects in an image, only pixels. The objects we see in them are not real artifacts but rather are inferences arising deep within the human cognitive system. When we see a "line" in a road map we average out the visual effect of thousands of pixels (some very ragged) and conclude we see a line. The computer sees only pixels of different colors.

People often would like to create a new, highly detailed drawing that can be used as a true vector GIS map, but they are put off by the major effort required to create such drawings. The idea of simply scanning in a paper map as an image and then waving some magic wand to make the computer recognize everything in the image and convert it to a vector drawing is a wonderful idea. But this is not yet reality. To date, computers cannot recognize objects in images and create vector equivalents in a way that automates the digitization of images.

If images have these drawbacks, why then does Manifold provide such a rich set of image manipulation capabilities? The reason is that images are very useful in GIS. People do undertake the effort of creating new vector maps despite the labor required. Images (satellite and aerial photos) are almost always the starting point. Not all drawings involve thousands of objects. Some drawings might show the layout of a farm or a few trails in a forest plot and could be created very rapidly by tracing over an image to create a new vector drawing. When assisted with smart editing tools (like those in Manifold) people can rapidly create very useful drawings from images in a matter of minutes. Images can also be used as beautiful backgrounds for vector layers in a map and can be used to provide realistic texture in 3D terrain simulations.

Raster data sets are also a very handy way of working with data that arises from remote sensors, such as those used for data acquisition from satellites. This and many other wonderful ways of using images are set forth throughout the rest of this documentation. For now, the new GIS user should trust us that images have their uses, but being a cheap substitute for vector drawings is not usually one of them.

**Key Idea:** The GIS landscape is still cluttered with living fossil ideas and products.

GIS as an industry is going through the same modernization that the computer hardware industry went through ten or fifteen years ago when aging, massive minicomputers from companies like DEC were replaced by personal computers from companies like Dell. Just as in the case of that hardware transition, the GIS software industry is now moving from a market populated by old-fashioned, overpriced legacy companies and products to a more modern market based on mainstream software ideas (such as those used within Manifold) that can deliver much greater capability at a much lower price.

However, just like minicomputer dinosaurs did not die off overnight the GIS market will take time to reach full modernization. Until that happens there will be many living fossils cluttering the GIS landscape. Such living fossils can range from the provision of government data sets in seriously obsolete formats to the tendency of conservative organizations, like some college faculties, to teach obsolete notions about GIS as if they were the latest thing. In fact, in some organizations GIS users may not even be aware that software has improved a lot in the last five or ten years.

Manifold users should therefore try to avoid being frustrated when they encounter GIS people talking about old-fashioned things as if they were the current state of the art. Be patient with such people and help them understand that software has evolved a lot in the mainstream and that GIS users can gain many benefits by using GIS software that leverages modern techniques and recent advances in mainstream computer software technology.
When learning more about GIS, keep in mind that GIS books and courses may be seriously obsolete. The reader should go forth and read all possible about GIS to gain deeper understanding and a better life. But at the same time, keep in mind that some "experts" and writers of books will falsely present obsolete GIS ideas and technology as contemporary stuff.

Some writers will also be so wrapped up in politically correct notions about "open" software that they will merrily ignore advances in the mainstream in order to advance obsolete notions about GIS which, although grievously behind the times, are "freeware." A useful touchstone in such cases is to see if the author gushes on about "GRASS," an obsolete program created by the US Army decades ago that is now freeware. If the author is excited about GRASS, that is almost proof positive it is time to turn the page and move on. On the other hand, if the author writes effectively about how to leverage modern software ideas like Microsoft's .NET framework, then that's a good indication the writer is in close contact with modern technology.

Note that manifold.net is not against learning all that can be learned from legacy software, or leveraging all that can be leveraged from "open" software: it is just that in doing so we should not give up the opportunity to leverage the mainstream and modern technology. It's a question of balance and inclusiveness, not religious mania for one thing or another. So, for example, it can be very cool to experiment with the open SourceForge .NET distribution of the Python language within Manifold even though this is not as "Microsoft" an approach as always using Visual Basic scripting.

As mentioned in the beginning of this topic, for many years GIS lagged far behind progress in mainstream computer markets. Until the PC revolution came to GIS, the GIS market was dominated by very old products from very old companies. Many of the books written about GIS and many college courses teaching GIS really do not teach GIS so much as they teach the ins and outs of working with specific, seriously obsolete software packages. It is as if one signed up for a course on computers and ended up in a college course that taught how to program IBM 370 mainframes in COBOL using punched paper cards.

Even in cases where books and courses do not take a "trade school" approach to teaching a specific obsolete package, the authors may be under the influence of obsolete technical ideas. Older generations of GIS programs required a massive amount of user activity to help the program do what in modern times we do with a point and click of the mouse. Quite a few people still teach GIS as if one still must program the GIS application using some command line interface to do elementary tasks. Avoid such teachers.

A handy rule of thumb to judge the modernity of GIS teachers is to assess the degree to which they focus on Microsoft Windows operating systems. If they are fluent in Windows concepts such as .NET and illustrate their lectures with practical examples of data interchange via OLE DB, ODBC, ADO.NET or other Windows concepts, it is likely that they have a contemporary view of software in general and GIS software in particular. If they start a lecture by delving into the intricacies of dead languages like AML or Avenue, they should be avoided.

We realize this comment may annoy those readers who favor non-Windows systems, but it is a useful rule of thumb nonetheless. As a software ecosystem, Microsoft Windows features intense competition between tens of thousands of applications. Windows markets include hundreds of millions of buyers who have extremely high expectations for maximum performance at rock bottom price.

The result is that Windows markets have featured faster evolution of user interfaces and many more new ideas for interactive software than any other market. Advances in user interfaces are one of the key measures of the modernity of a GIS, so expertise in Windows is a useful measure of the modernity of a GIS (and a GIS instructor) because only very modern user interfaces have a chance in the Windows ecosystem.

For additional radical manifold.net perspectives on GIS, see the Essays topics. To continue this introduction to Manifold System, proceed to the main Key Ideas topic.
Key Ideas

This topic assumes the reader is familiar with basic GIS concepts. If you are new to GIS, please read the Key Ideas in GIS topic before proceeding with this topic.

**Key Idea: Manifold requires Microsoft Windows fluency.**

Manifold is a Microsoft Windows application through and through. Success with Manifold requires fluency with Windows. This is good news, because Windows expertise built up with Manifold will serve you in many other applications as well.

Windows includes many helpful techniques and "power moves" to make life easier for people who work with computers. If you have not yet taken the time to master Windows, now is the time to do so. Most Windows power moves are used within Manifold. For example, when highlighting records in a table to quickly highlight a group of records simply click on the first record to highlight it and then SHIFT-click on the last record and this last record as well as all the records in between will be highlighted.

While some Windows power moves are a matter of convenience, other power moves (such as those involving Clipboard Copy and Paste As) are so important within Manifold that fluency with them is mandatory for success with Manifold. Read the Windows topic to learn essential Windows nomenclature and techniques as they are applied to Manifold.

Because this documentation uses exclusively Windows nomenclature, it is **absolutely essential** that users new to Windows read the topic to learn what terms like "the focus" mean. The Windows topic also describes essential Manifold power moves used by expert Manifold users, so it should be read even if you are already a Windows power user.

**Key Idea: Manifold uses context sensitive windows, toolbars and panes.**

The different windows and panes that are may be opened in Manifold all maintain their own contexts. Whenever we click on a window, Manifold knows what type of window it is and will automatically enable and disable menus and controls as appropriate for that type of window. Any control panes that are open will automatically switch their contents to show the correct information and settings for the active window. Toolbar buttons will appear and disappear as appropriate for the type of window that is active. Toolbar buttons that cannot be used in the current context will be disabled. For more on the philosophy behind Manifold's user interface, see the essay User Interface Design in the Appendices.

**Key Idea: Projects are the “documents” of Manifold.**

The key documents for Manifold are projects, saved as a single .map file. These are analogous to the .doc files that are the documents for Microsoft Word, or the .mdb databases that are the documents for Access or the .xls spreadsheets that are the documents for Excel. Whenever we do a File - New or File - Open or File - Save command in Manifold we are creating or opening or saving a .map project file.

Projects contain components. Components are items such as drawings, images, tables and maps within the project. The minimum series of steps to create a map in Manifold is to:

- Create a project.
- Import or create a drawing or image.
- Create a map based on that drawing or image.

All the components of a project are kept in one file, the .map format project file (".map" = manifold project).

We must resist the impulse to imagine that the different components listed in the project pane are different files. They may have been imported from different files but now that they are in the project they are all stored as one file. This makes it easy to transfer a project from one place to another without risk that some important part will be left behind.

There are three exceptions to above rule of keeping all data in the project file. The first exception is that tables can be either imported into the project or they can be left as external tables that are dynamically linked into the
The second exception is that linked drawings are automatically created on the fly from an external geocoded database table.

The third exception is the use of Enterprise Edition to save components within Enterprise servers. Enterprise Edition allows control over whether shared components are cached or not cached. If they are not cached, they will take up no space in the local project but rather will be stored only on the Enterprise server.

When working with a project that uses external tables or connections to Enterprise servers we take on, of course, the responsibility of correctly maintaining links if we move the project to a different computer system.

**Key Idea:** The differences between drawings and images.

**Drawings** are things like blueprints, CAD drawings and GIS vector maps. They are made up of precise points, lines and areas between exact coordinates that retain an exact, crisp appearance no matter what the zoom level. The points, lines and areas in drawings are objects that may be selected and manipulated as distinct entities. The objects in drawings are linked to database records so that the drawing can be used as a visual user interface to grab and display data in the database. Almost all GIS maps will be imported into Manifold as drawings.

**Images** are things like photographs, scanned documents, or raster satellite scans. Images are made up of pixels. Zooming into an image results in a less crisp view as the individual pixels blossom into view. What appears to be a sharp line or point at one zoom level will drastically change appearance when we zoom in: at high magnification it will be seen as a line-shaped or round cloud of pixels. There are no "objects" in images - only pixels. What appear to be distinct entities in an image is simply the result of human cognitive perception that averages many pixels into some recognizable thing.

To use a word processing analogy, a drawing is like a text in a Microsoft word document made up of individual letter "objects." By changing the font used in Word, we can make the same letter six points in size or 128 points in size without any loss of meaning. Changing the formatting of the letter does not change the letter from, say, an A to a B.

In contrast, an image is like a .gif or .jpg screen shot of a text document. If you've ever downloaded an image from a web site that includes writing, you know that such an image cannot be edited to change the font. One cannot highlight a "word" in a .gif image and copy it into Microsoft Word. If you open the image in Microsoft Paint or Microsoft Photo Editor, you know that the text it apparently contains is nothing more than a cloud of pixels in letter shapes. The appearance of text that we see is simply the result of how our eyes and brains average out the pixels to recognize letter shapes. We cannot "grab" the text and change the font, because there is no text really there…. it's just a cloud of pixels in letter shapes.

There is a similar difference between vector drawings used in maps and raster images used to show maps. If we have a "vector map" - a drawing - we can grab and move and change the elements of the map with great precision just like we can highlight a line of text in a Word document and change the font size. That cannot be done with images. Because it is easy to scan a paper map than it is to create a precise vector map, some older types of GIS programs use maps that are scanned raster images. These are easy to make at first (just scan a paper map in the scanner) but they cannot later be changed in precise ways. We can no more "grab" a road line in such a map and change its size than we can grab the pixels of a .gif that shows letters and change their font.

Manifold includes extensive raster capabilities that may be used to display and manipulate maps that are made up of scanned raster images. However, it is important to understand the strong points and weak points of using vector drawings and raster images in different situations. Very often a cool map will combine both drawing and image layers. For example, an aerial photograph image might be used as a background to provide context for a vector drawing of a road system.

See the individual topics on Drawings and Images for more detail on the differences between the two and how they are used.

**Key Idea:** Formatting determines the appearance of drawings.

Images look the way they do based on the colors and arrangement of their pixels. Drawings are different, because they are made of lines and points and areas that look like abstract figures until you tell them what they are supposed to look like.

Lines in drawings will look like plain, thin, black lines by default. We can format them to specify different colors, thickness or line style to achieve different visual effects. For example, we could format a line to look like a path of alternating red and black boxes. Points look like small dots by default. We can increase their size or change
their color or use different styles (including complex icons) to represent points. By default, areas look like gray shapes bordered by a thin boundary of darker gray.

Changing the formatting of objects in maps is analogous to changing the font formatting of text in Microsoft Word. In Word, we highlight text to select it and then we can apply formatting. In Manifold, formatting is applied throughout an entire drawing at once. To create a map that shows different objects in different formats we can stack many drawings as layers. Each drawing can use whatever format is desired so that the map creates the overall effect desired.

Formatting is easy and fun but it requires understanding that the formatting applied to the drawing is a different concept than changing the actual contents of the drawing. It’s like formatting a text in Microsoft Word: changing the font formatting of a text in Word can radically change the visual appearance of that text but the content (the actual words) stays the same no matter what formatting is applied.

A refinement: Drawings may also use thematic formatting, where the colors or styles of objects are automatically based on the data they contain. For example, a map showing provinces might be formatted so that the provinces are automatically colored by their per-capita income taken from a table of demographic information for each province.

Key Idea: Selection

Just about everything we do in Manifold involves selecting something first and then applying an operation to that something. We can select some pixels in an image, some objects in a drawing, or some records in a database. Manifold has a standard, uniform way of making selections and then combining them through selection modes and selection commands. It’s a simple idea that once understood opens the door to great power. How to use selection commands in Manifold is a key idea that must be mastered. See the Selections topic for the essential introduction.

There are many different ways to make selections in Manifold. Visual selection commands allow us to select what we see by clicking on something with a mouse or by drawing a mouse box that encloses items to be selected. If we wish to select items based on the data associated with them we can use SQL in a query to select something with industry-standard Structured Query Language (SQL) based on the data attributes. We can click on items we see in a chart or we can highlight individual records in a database table. We can even use solvers or programs to select items. Most importantly, we can combine these different methods to get exactly the right selection desired using the most convenient view of the data. So, we might select all the counties in a particular region with a mouse box and then view those counties in a database table so we can further select only those with the largest populations based on the values of a population field.

Visual selection uses selection filter buttons to restrict what is selected by the mouse. There are four filter buttons, one each for points, lines, areas and pixels. The filter buttons allow or disallow selection of that type of item. If the area and line buttons are pushed in but points and pixels are not pushed in, then any mouse motion will select only areas and lines and will ignore points and pixels.

Selection mode buttons specify how any new items picked out are combined with any previous selection. We can use selection modes to replace the previous selection, or to add or subtract items from the selection. More sophisticated modes such as Invert and Select Intersect give fine control in complex maps.

The Selections dialog provides a means of saving particular selections for each window. After spending much time to pick out exactly the right set of objects or pixels, it is nice to be able to save this selection so that if we ever need to work with this set of items again we can fetch it.

Key Idea: Maps and Layers

Maps are project components that are made up of drawings and images stacked in layers. Layers are a very cool and easy way of “stacking” drawings and images on top of each other to create exactly the visual effect desired in the final map. Clicking open a map shows it in a map window.

Double clicking on a drawing opens it in a drawing window, and double clicking on an image opens it in an image window. There is little difference between a drawing or image window and a map window when we have only one drawing or image in our map. Why bother to create a map if we will look at only one drawing?
some of them and move them to a new layer (this automatically creates a new drawing with the name of the new layer and moves the objects into that layer).

For virtually all projects it is best to work in a map window. This will encourage good organization through the use of layers.

**Key Idea:** Databases and Maps

Probably the single most distinctive idea in a GIS is the notion that the visual objects in a drawing can be linked to records in a database. If we have a drawing of provinces and a database of demographic records for each province, for example, we can link the picture of each province in the drawing with the database record for that province. We can then grab records in the database by pointing and clicking on objects in the map. This is a very simple idea of immense power.

Most Windows power users have a lot of experience with databases so it’s quite likely Manifold users already understand basic ideas like records, tables, data fields and so on. Using databases with maps involves a few additional ideas that are straightforward extensions. To support work with database information, Manifold includes many database management features (such as SQL) as well as some powerful database management facilities that are not available in ordinary database products.

Every drawing in Manifold has a database table linked to it. Each object in the drawing corresponds to a record (one row) in the database table. Every field in the database will be a column in the database table. Tables linked to drawings will have at least one field, the object ID field, which provides an identification number for each object in the drawing. Tables can contain other fields as well. For example, a drawing showing cities as points could have a Name and Population field in the table that gives the name of each city and its population.

A drawing’s table can be used like any database table. For example, it can be used to form relationships with other tables, including tables that are saved outside of Manifold in external database providers.

Manifold can also work with database tables that are not linked to objects in a drawing. Manifold provides a vast array of capabilities that may be used to manipulate and analyze database tables of all kinds.

**Key Idea:** Use the Transform toolbar for simple tasks.

The Transform toolbar provides a vast number of operators that may be applied to drawings, images or tables in a one-click way. Functions in the Transform toolbar either have no parameters or only one parameter that need be specified. This allows them to be fitted into a simple toolbar with three option boxes. The transform toolbar will automatically be configured with operators that make sense for the active window. To use the transform toolbar, choose an operator and press **Apply**. If there is no selection, operators will be applied to the entire component. If a selection exists, most operators will be applied only to the selection.

**Key Idea:** Use Layouts for printing

Although Manifold can print directly from any component window, we can achieve finer control over the print job by first creating a print layout and then printing the layout. Layouts provide a highly WYSIWYG (What You See Is What You Get) setting within which we can compose a print job while seeing exactly how it will look on the printed page.

**For More Advanced Users**

- **Projections and Coordinate Systems** - There’s no way around this one! If we want to combine images with drawings in a map we need to know some basics about coordinate systems used with maps and drawings. Sooner or later all intermediate or advanced mapping will involve geographic projections as well. In particular, importing maps often requires knowledge of projections because maps found on Internet are often published as projected maps using "dumb" formats that are poorly suited for publication of projected maps. If we want to be able to use such maps we will need skill with projections just to get around the foolishness of the publisher. (It's as if we needed to learn Latin to download MP3's from the net…) The best course when it is time to learn about projections is to dig right in and do the reading and experimentation that is required.

- **Customization** - Working with the Manifold user interface to change it requires understanding how to use add-ins and scripts to add new commands to the Custom toolbar. Users may also add their own point styles, palettes, custom units of measure and even custom ellipsoids, datums and projection presets.
• **Programming** - Manifold may be programmed within the system using Active Columns, scripts and script forms using ActiveX scripting, or we can program Manifold from external programs written in standard Microsoft languages such as Visual Basic. Manifold includes all software necessary for programming within the system using scripting languages such as Visual Basic scripting or JavaScript. Programming from an external development environment, of course, requires that you purchase and install the external development environment. The Manifold development team prefers working with Visual Basic or Visual C++ within Microsoft Visual Studio, but any Microsoft compatible development environment may be used.

• **Internet Map Server (IMS)** - Use Manifold IMS to publish your Manifold work to live, interactive web sites. Manifold IMS works together with Microsoft Internet Information Server to create spectacular web sites in Microsoft web-serving environments. Simple web sites using default templates can be created without any programming at all. More advanced webmasters, of course, have full freedom to edit the .asp files created by Manifold IMS or to write their own for fully custom applications. Because Microsoft IIS is delivered free on most Windows systems, it is quite likely that most Manifold users have the ability to create a web site on their machines. Such a web site can be viewed throughout the local network and, if the machine is connected to the Internet via a full time link (increasingly common in an age of DSL, cable and other broadband connections) can be published to the entire world. Every Manifold user should take the time to put up an IMS web site if possible. It's remarkably easy to do, very impressive to visitors and a great way to share your GIS work with others.
Introduction

Windows
This topic provides a refresher course in Microsoft Windows "power moves" used within Manifold. It also discusses the standard Windows nomenclature we use throughout the documentation. Finally, it introduces special Manifold power moves used within Windows.

Very Important: Before starting Manifold, go to the Windows Control Panel Display dialog Effects tab and make sure that the Show window contents while dragging check box is not checked. In Windows XP this option is found in the Control Panel's Appearances and Themes - Display choice under the Appearance tab by pressing the Effects button. Checking this box will greatly slow down the system when displaying complex maps and images because it forces a refresh of the window contents with each minor change in mouse position while dragging the window.

Windows Display Settings
Manifold uses many toolbars, some of which will not fit in the default arrangement in lower resolution display settings such as 800 x 600 or 1024 x 768. Rearrange toolbars into multiple horizontal rows or undock some toolbars when using lower resolution display settings. If you cannot find a particular toolbar button, check to make sure the entire toolbar is visible.

Although it's possible to work effectively with a 1024 x 768 resolution, a monitor and video card combination that can show 1280 x 1024 resolution or higher is recommended. You will be able to work faster and more conveniently if there is plenty of room for work area in addition to toolbars and panes a display resolution.

Windows Explorer Settings
Microsoft Windows Explorer by default uses settings that hide the three-letter extension that is part of most Windows files. For example, a file called readme.txt will be shown by default just as readme and will appear with an icon associated with Notepad.

Since the only way to identify many different GIS data files is by the three-letter extension at the end of the file (such as .shp, .mif, .mdb, etc.), it is a good idea to always display the extension. To do so, in Windows Explorer choose Tools - Folder Options and in the View tab uncheck the Hide file extensions for known file types option. This will cause Windows Explorer to always show the three-letter extension for all files.

Experienced users will also often change the setting in the View menu of Windows Explorer to Details so that they can immediately see the size and date of each file. This setting provides a more efficient view than a window full of large icons. The Details setting is also very handy when seeking the most recent file in a folder, because one can then simply click on the Modified column head to sort files in the folder by date of most recent modification.

Windows Files Copied from DVD are Read-only
When copying a file from a DVD to a folder on the hard disk, Windows will create the file on hard disk as a read-only file. To work with these files as one might like in Manifold take a moment to uncheck the read-only attribute. To do this, right click on the file in Windows Explorer, choose Properties and uncheck the Read-only box in the Attributes section. Press OK.

The Active Window
Manifold launches a main window that becomes populated with subsidiary windows as we open projects and components of projects. The illustration above shows the project pane and an image window within the main window as well as a floating toolbar. Toolbars can be floating or “docked” into position into the frame of the main window.

Windows and undocked toolbars have title bars, the blue colored region at the top with the name of the window or toolbar. Move windows about by clicking on their title bars and dragging.

Only one window at a time is the active window in Microsoft Windows. Clicking into a window or touching it in some way activates it to make it the active window. The active window is also said in some contexts to have the focus. When a window is active its title bar changes to the highlighted color specified for active windows in the Display appearance settings specified in your Windows Control Panel. The active window is always on top of other windows.

In addition to the change in title bar color, windows will often have subtle visual cues to help indicate when they do not have the focus. For example, in Manifold text windows such as comments, query or script windows that may have text selected in them with the usual “white on black” block indication used in Windows that the text is selected will have the dark background used to indicate selection faded to gray when the window does not have the focus. This makes it easier to immediately see the active window when many text windows are open.

Manifold windows are context sensitive. The main menu and toolbar controls will change to support use of whatever window is active. Manifold toolbars and toolbar buttons are designed to preserve commonality of function as much as is possible, so that when the focus changes from a drawing to an image window the selection buttons will retain their appearance. However, their function will change in subtle ways to support the slight differences in selection operations between objects in drawings and pixels in images.

Some applications will use a window’s title bar to report information about that window. Manifold does not do this because there is usually more information we like to see than will fit in the title bar. Instead, Manifold uses the main Manifold window’s status bar at the bottom of the screen to show information relevant to the active window.

**Resizing Windows**

Windows may be resized by clicking and dragging on their borders. Clicking and dragging on a window’s corner will resize it both vertically and horizontally at once.

Each window has three small buttons in the upper right hand corner. These are:
Minimize 
Minimize this window into a floating title bar.

Maximize 
Maximize this window to fill the main window.

Close 
Close this window.

Minimizing a window converts it to a small floating title bar positioned near the bottom of the main window. This is a useful method of getting a window temporarily out of the way when working with many windows at once. Minimized windows may be dragged about by their title bars to a new position.

The restore button will appear in the minimized window’s title bar. Click this button to restore the window to its previous size.

Maximizing a window expands it to fill the main Manifold window. The title bar will be combined with the title bar of the main window. We can also maximize a window by double clicking on the title bar. Note the default checkerboard background of an image window that appears wherever the image has no pixels. See the Layers pane topic for info on the background.
Maximized windows have their restore control buttons just below the window buttons of the main Manifold window.

Click on the **restore** button to restore the maximized window to its previous size.

When maximizing one window, Microsoft Windows assumes we want all windows to be maximized. Therefore, if we have several windows open and one of them is maximized and then we switch to another window (either by closing the “top” window or by using the **Windows** menu to switch to a different window) we will see that all of the other windows are maximized as well.

**Autohide / Popup Panes**

Panes used for controls can be minimized when docked so that they autohide themselves when not in use and automatically pop open when the mouse cursor moves over their tab. Virtually every experienced Manifold user employs this feature to maximize the amount of screen area available for working with component windows. See the **View - Panes** topic for details.

**Full Screen Mode**

Choose **View - Full Screen** or press the **F11** key to expand the main Manifold window to full screen mode, using the entire monitor screen and eliminating the title bar and menu bar. If toolbars are hidden using **Tools - Customize** and a window is maximized the result is a “full screen” display. Press **F11** to get out of full screen mode.

**Arranging Windows**

The **Window menu** allows us to arrange windows within the main Manifold window.

- **New Window** - Enabled when a component window is activated. Open a new window into that component.
- **Close** - Close the active window.
- **Close All** - Close all windows.
- **Next** - Activate the next window in the stack.
- **Previous** - Activate the previous window in the stack.
- **Cascade** - Arrange windows neatly in front of each other and offset slightly.
- **Tile Horizontally** - Arrange windows next to each other left to right and resize so they fill the main window.
- **Tile Vertically** - Arrange windows next to each other top to bottom and resize so they fill the main window.
- **Windows...** - Open a dialog that lists all windows.

See the Windows Dialog topic for additional info.

**Panes**

**Panes** are a type of window that can be left “always on top” or docked into the left or right edges of the main Manifold window. Panes are used to display information or controls that are always to be kept in view while work goes on in other windows. Panes may be repositioned by dragging their title bar to a new position. Panes may be resized as well.
Manifold has several panes, such as the project pane and the layers pane, that are important parts of the user interface and which are normally left open. Panes can be turned on/off using the View - Panes menu or with keyboard shortcuts like SHIFT-ALT-P (open/close the Project pane).

See the View - Panes topic for a list of panes and to open and close panes. Panes may also be closed by clicking the X box in their caption bars. Familiarity with panes and their use is a mandatory skill within Manifold.

Note: The word "panes" can also be sub-areas of main windows that might be used to keep some information in view while another part of a window shows a changing display.

Toolbars

Manifold duplicates many menu commands in the form of toolbar buttons. Manifold command buttons all have Tool Tips, which are quick labels that pop open if the mouse cursor lingers on the button. Toolbars may be turned ON and OFF under the Tools - Customize menu. A right mouse click on the main menu bar calls up a shortcut to the Tools - Customize menu.

Individual toolbars may be dragged to different locations on the display. They may be "docked" into the upper or lower screen edges or left "floating" in the middle of the display. Move a docked toolbar by dragging the 3D ridge at the left or top of the toolbar. Move a floating toolbar as you would any window, by dragging its title bar. Use a CTRL-drag to drag a toolbar past the normally docked position.

Manifold toolbar buttons are context sensitive and are not enabled unless they can be applied. For example, unless we are working on an image the Select Pixels selection mode button will not be enabled in the Selection toolbar.

The toolbar above is in its undocked form, with a title bar. In docked form, toolbars do not have a title bar. Toolbars may be undocked by dragging them (using the small 3D ridge at the left side or top of the toolbar) into the main Manifold window pane.

To quickly see which toolbar is which, use Tools - Customize to turn it ON and OFF. As the toolbar turns ON and OFF it will be obvious which it is.

Docking and Undocking

Toolbars and panes can be docked into the main Manifold Window. Because toolbars are very wide but not tall, they can only be docked into the upper or lower edges of the main window. Because panes are normally taller than they are wide, they can be docked only into the left or right edges of the main window. A docked pane will expand in height to fill (or share) the available vertical space at the edge of the main window.

Toolbars and panes may be docked into the top, bottom, left or right margins of the main Manifold window. Most toolbars are docked by default and most panes are not docked by default.

To move a docked toolbar, click and drag on the toolbar's move handle. To undock a toolbar, move it into the main work area.
An undocked toolbar may be docked by dragging and dropping it into the desired position in the main window's margins.

When a pane or toolbar is brought near to or on top of a margin using a click and drag, it will automatically dock itself. To override this, use a CTRL click and drag. This will allow you to drag a pane or toolbar over top of a toolbar row or near to a window margin without the pane or toolbar docking itself.

To undock a docked pane, click and drag it from the docked position into the main Manifold window area. Click and drag on the small 3D ridge at the left side of the docked pane to move it.

To undock a docked pane and drop it near the margin, use a CTRL click and drag onto the title bar.

Drag it to a position near the border while holding down the CTRL key. The CTRL tells Manifold not to dock the pane even though it is near the border.
Release the mouse at the desired position of the undocked pane. The illustrations above show a greatly minimized main Manifold window. In real life we would use a reasonably large computer display with high resolution so we can display many controls and panes while leaving room for practically sized working windows.

Panes may also be resized by dragging their horizontal or vertical borders. When panes are docked together, the borders between them may be dragged to resize them relative to each other.

**Mouse Moves and Context Menus**

A **left click** or left click and drag of the mouse is normally an "action" command that does something. A **right click** is normally an "information" command that requests information or a choice of options. Whenever we simply say "click" the mouse, we always mean a **left click**.

Right clicking into windows or onto parts of windows or other controls will often call up a **context menu** of available choices. For example, right clicking onto a layer tab in a map window will pop open a context menu of commands that may be applied to that layer. Right clicking onto a column head in a table will pop open a context menu of commands that may be applied to table columns, such **Sort Ascending** and **Sort Descending**.

**Using the Scroll Wheel on Wheel Mice**

Some mice are equipped with a scroll wheel, normally positioned between the two buttons of a mouse. The scroll wheel may be used in Manifold to zoom in and out of visual components like drawings or images and to scroll up and down in text components like tables.

Using a scroll wheel in a drawing, image, labels, layout, map, profile, surface or theme window zooms in and out. Holding the **Ctrl** button down on the keyboard while using the scroll wheel will force zooming near the center of the window.

When zooming, each zoom step will be made using the **Zoom magnification factor** specified in the Tools - Options - User Interface dialog. This factor specifies the change in magnification to use when zooming in or out in the range from **1.1** to **4.0**. The default value is **2.0**.

Using a scroll wheel with text components like tables or comments will scroll up and down through the text, just like moving a scroll bar up and down.

**Note**: When Manifold is used as an ActiveX control within other applications the scroll wheel will still zoom in and out.

**Windows Dialog Controls**

The various buttons, edit boxes, menus and other items that appear in a Windows dialog or other window are called **controls**.
**Color wells** are squares that provide a color choice. Click into a color well to choose it, for example, in the main Windows color dialog.

Color wells also occur in toolbars.

Clicking into a color well in a toolbar will usually call up the Color dialog to enable setting that color. Manifold’s toolbar color palettes will provide an array of colors where the first row presents variations of the current color. Choosing More calls up the classic Windows color setting dialog.

**Check boxes** are simply on/off controls. Click the box to turn that item on/off. A special type of check box is an option button or radio button, which is a small round control with a dot in the middle. Option buttons are used in groups where only one option button of the group may be checked. [Hence the slang term "radio button", since just like the channel choice buttons in old car radios pushing one button in pushes all other buttons out.]

**Text boxes** or Edit boxes allow users to enter or choose data. Edit boxes will often have lists that provide a choice of allowed options, in which case they are known as a List box. Click on the down arrow button to see the list and then click on the choice desired. Some list boxes will have so many choices that they will have a scroll bar that may be used to scroll through the list. Dropdown lists that appear are also known as drop down or pull down menus.

When navigating a long list of choices in a list box entering the first letter of a desired choice will scroll the list to the first entry beginning with that letter.
For example, the transform operators list box in the transform toolbar for drawings has many choices. With the mouse positioned in the list if we would like to scroll directly to the **Union** operator we can enter the "u" character at the keyboard.

The list will immediately scroll to **Union** and highlight that operator. We can then press **Enter** to choose that operator or click on it with the mouse to choose it.

Some edit boxes allow either choosing from a list or they will also accept a value entered by the user by clicking and entering text into the box. Such edit boxes are also known as **combo boxes**. If we can click into an edit box in Manifold, we know that we are not restricted to only those choices presented in the pull down list.

Edit boxes are often associated with **slider** controls in image commands. Moving the slider back and forth will automatically change the value in the edit box. Alternately, we can click into the edit box and enter a number directly.

Once clicked with a mouse to become activated, a slider bar can also be moved left or right using the **Left Arrow** and **Right Arrow** keys on the keyboard. This is a very handy method when working interactively with a previewed control. For example, when setting the level of water in a terrain window for waterline effects, we would open the **Terrain - Water** dialog, check the **Preview** box, click on the **Level** slider and then use the left and right arrow keys to move the slider up and down.

**Spin buttons** or **UpDown** controls are small up and down arrows inside some edit boxes. Clicking on the up arrow increases the value in the edit box by a pre-designated amount. Clicking on the down arrow decreases the value in the edit box. Most edit boxes with spin buttons will also allow direct user entry of the value by clicking into the box without using the spin buttons.

Once clicked with a mouse to become activated, the value in an edit box with spin buttons can also be increased or decreased using the **Up Arrow** and **Down Arrow** keys on the keyboard.

**Note:** Manifold's built-in programming environment allows us to create our own dialogs and forms using the above controls and many, many more. See the Form Controls topic for information.
When working with a Windows dialog, only one control at a time in that dialog will have the **focus**. The control that has the focus will normally be highlighted with a fine dotted outline or other indication that it has the focus.

![Yes No](image)

Clicking into an edit box will move the focus to that edit box, which is shown with the cursor in the edit box. Pressing **Enter** on the keyboard applies the action of that control. For example, in the illustration above the **Yes** command button has the focus, so if we press **Enter** on the keyboard it is as if we pressed the **Yes** button with the mouse.

Clicking the **Tab** key on the keyboard will move the focus between controls in the dialog using whatever **tab order** the designer of the dialog specified. Many windows dialogs will start out with the focus on the **OK** key or other "yes, do it" key in the dialog. This makes it easy for people to respond to dialogs by simply pressing **Enter** on the keyboard if they agree with the default settings shown.

The **ESC** ("escape") key on the keyboard is used to exit or abandon many mouse modes or commands in windows. The **ALT-Tab** key combination may also be pressed as an equivalent to **ESC** if some program or information dialog (such as a "download complete" when downloading files in background) has snatched context away from the Manifold window.

Pressing the **ESC** key will also escape from any tool mode, such as zoom tool or selection tool and will restore default "smart mouse" selection.

**Changing Values**

Many Windows dialogs will provide lists of items (files, components, properties, parameters, etc.) with a name or value for each item.

![Map 2](image)

Clicking on the item will usually highlight it.

![Map 3](image)

A rapid double click will often open the item (if it may be opened), whereas clicking once to highlight the item, pausing, and then clicking it again will open it for editing. For example, we can rename "Map2" to "Globe".

![Globe](image)

We enter the characters "Globe" and then press **Enter**. If we wished to abandon the edit we could press the **Esc** key.

![Globe](image)

The above edit sequence may often be accomplished by right clicking on the item and choosing **Rename** from the context menu that pops open.

**Jumping Through Long Lists in Combo Boxes**

When choosing items from a long list in a combo box, Windows allows us to enter a keyboard character and the list cursor will jump to the first item in the list that begins with that letter. Entering more than one character quickly will jump to the first match in alphabetical order to the text entered.
Windows uses a timer to tell the difference between typing single letters and typing a group of letters to be interpreted together as a word. For example, using the Query toolbar’s Operations combo box, if we click into the combo and press M the selected operator will change to Maximum, the first choice in the Operator list that begins with the letter M. If we wait a few seconds and press E on the keyboard the selection will change to Equal to, the first choice that begins with the letter E.

If we wait a few seconds more and then rapidly type ME the selection will jump to Median, the first choice that begins with the letters ME. Because the letters were typed rapidly together Windows knows we want them to be interpreted together as a text string and not as a desire to first jump to the Maximum choice and then to jump to the Equal choice. If we had paused between the two letters Windows would have understood them to be separate choices.

Conveniences in Windows Dialogs

Manifold provides a rich set of Windows features in most dialogs. The illustration shows a Windows 2000 dialog.

For example, the illustration above shows the File - Import - Drawing dialog set to import SHP files. The four icons in the left-hand pane of the dialog allow convenient jumps to different places in the file system.

The History icon is especially convenient because it provides shortcuts to folders and files previously visited. If we import files from two or three different folders (which could be located on different machines) the History icon lets us jump directly to any one of them.

Most Windows dialogs in Manifold are “live” Windows Explorer sessions as well. Right click into the pane to call of a list of commands. For example, within the Import dialog seen above we can rename files that are displayed, create new folders, copy and paste files and so on. [In the example above we see an ESRI .shp file. “Shapefiles” are actually a set of three files that have the same name but ending with a .shp, .dbf and .shx extension. If we are tempted to copy the .shp file seen to have a backup copy we would have to remember to
copy the .dbf and .shx files as well. One reason Manifold saves files to a single .map file is to avoid such opportunities for errors].

The Files of type box provides a filter that normally shows only those files ending with the specified extension. This is a useful way of seeing only those files that end with a .shp extension.

The illustration above shows a Windows 2000 dialog. Because Manifold uses the facilities of the host operating system to provide standard dialogs with standard controls, the appearance of dialogs will vary to match prevailing standards in whatever version of Windows is in use. Some features provided by Windows 2000 or XP will not be available in earlier Windows versions.

The Windows Clipboard

Windows operating systems provide a Clipboard, which is a temporary storage location to which almost anything in Windows can be copied. The Clipboard is common to all Windows applications. Expert use of Windows almost always involves the Clipboard.

Whenever something is copied using Edit - Copy or a Copy command in a standard Windows context menu the data that represents that something is copied to temporary system storage within the Clipboard. Using Edit - Paste or a Paste command from a context menu will paste whatever is in the Clipboard into the active window.

Applications programs vary in their sophistication and ability to use the Clipboard. Virtually all applications can handle copying of text to the Clipboard and pasting of text from the Clipboard. More sophisticated applications are able to copy more complex data, such as images, and to accept pastes of complex data as well.

Within Manifold we use the Clipboard for a very wide variety of applications. Virtually everything within Manifold is Clipboard-enabled. Use the Clipboard for simple things, such as copying the name of a file from Windows explorer and pasting it into a new name for a project or component. Use the Clipboard for more complex work such as copying geographic objects from one drawing and pasting them into another, or for copying pixels from one image and pasting them into another.

There may be some limitations to the universal use of Copy and Paste caused by incompatibilities between the source and destination. For example, when using Copy to copy multiple layers from a map one can only Paste back into a map.

Copy and Paste As

Manifold uses standard Windows Copy and Paste As to convert components. For example, the way we create a drawing of points from a geocoded table is to Copy the table in the project pane and then Paste As a drawing. See the Copy and Paste As topic for additional information.

Drag and Drop

"Drag and Drop" is Windows jargon for a simple user interface method that's widely used to copy things or to pick items from one window and insert them into another.

To drag and drop an item:

1. Make sure that both the source window and the destination window are open.
2. Click on the source window to make it active. Position the mouse cursor over the desired item.
3. Press down on the left mouse button and hold it down.
4. Without releasing the left mouse button, move the mouse cursor into the destination window. This is the "drag."
5. Release the left mouse button. This is the "drop."

Drag and drop is used within Manifold in several ways:

- Add components to a map by dragging and dropping them from the project pane into the map. If a map window is open and we would like to add another component from the project pane into the map as an additional map layer, we can simply click on the component’s name in the project pane and then drag
and drop it anywhere in the map window. We don't have to "drop" it onto the layer tabs. We can drop it anywhere in the map window. This component will then be opened as part of the map as a new layer within the map.

- Copy or move controls between forms components. Dragging and dropping a control from one form into another form will copy or move that control.

- Copy items between two different instances of Manifold. Launch Manifold once from the Start button and then launch it again. Resize both instances of Manifold so you can see them on screen. Items can now be dragged and dropped between the two different project panes.

- Adding components to a print layout by dragging and dropping them from the project pane into the layout.

Hot Scroll

Manifold does not use scroll bars in image, drawing, labels, surface or map windows because such windows are limitless in extent. Instead, Manifold uses hot scroll commands using the ALT keyboard button and right mouse clicks. ALT -right click near the edge of a window to move in that direction. Remember ALT as a mnemonic for the "A" in "Arrow keys."

Using ALT or SHIFT-ALT or CTRL-ALT with either a right click or with keyboard arrow keys has the following effect:

- **ALT- (right click)**: Scroll left, right, up, down or in diagonal directions in direction clicked by mouse.
- **ALT- (arrow key)**: Scroll left, right, up, down or in diagonal directions in direction of the keyboard arrow key.
- **SHIFT-ALT- (right click)**: Long scroll. Scroll in greater jumps at each step.
- **SHIFT-ALT- (arrow key)**: Long scroll. Scroll in greater jumps at each step.
- **CTRL-ALT- (right click)**: Jump to top / bottom / left / right edge of displayed data.

Wheel mice are supported in Manifold in cases where scroll bars appear. For example, tables showing horizontal scroll bars will scroll when the wheel is rolled on a wheel mouse.

ALT is also used to extend selections and Clipboard operations within maps to all visible layers (ALT is a mnemonic for "All").

Interactive Repositioning Methods

If the mouse is not occupied in a command mode, we may interactively reposition layers in maps using the following mouse and keyboard commands. To use these commands, the layer being repositioned must use the same projection as the map. See Repositioning Layers.

- **CTRL dragging**: Pressing the CTRL key while clicking and dragging with the mouse will reposition the layer. The method is similar to using the Grabber tool but whereas the grabber tool pans the view, using CTRL dragging moves the actual layer.
- **CTRL <arrow key>**: Pressing the CTRL key while pressing one of the keyboard arrow keys will move the layer in that direction.
- **SHIFT - CTRL <arrow key>**: Pressing both the SHIFT key and the CTRL key while pressing one of the keyboard arrow keys will move the layer in that direction in greater jumps.

Autoscroll

When the mouse is occupied in mouse motion operations such as selection, editing, painting, and so for we would at times like the window to automatically scroll when the mouse gets near the edge of the window. Manifold will
automatically autoscroll as the mouse nears the edge of a window. To turn this off, uncheck the **Autoscroll window on edit or selection operations** checkbox in **Tools - Options - User Interface**.

The sensitivity of autoscroll may also be set in **Tools - Options - User Interface** by specifying how close to the edge of the window the mouse can move before an autoscroll is triggered. The sensitivity is set by specifying the margin in pixels within which an autoscroll occurs.

**Additional Mouse Buttons**

Manifold component windows (such as a drawing, image or map window) will automatically work with standard commands issued by additional mouse buttons on mice so equipped, such as the IntelliMouse Explorer. For example, the fourth and fifth mouse buttons typically issue **Back** and **Forward** commands for views.

**Double Click Short Cuts**

Double clicking on a component in the Project pane will open the component in its own window.

Double clicking on legends, north arrows and scale bars will open the **Properties** dialog for those items. Double clicking on a legend, north arrow or scale bar within a print layout will open it for editing in a dialog.

Double clicking on an object in a drawing will launch your Internet browser using the contents of a **URL** field, if any, in the drawing’s table for that object. If there is no **URL** field or if it is not of type "URL" or if it is empty the **Fields** dialog will be popped open.

**CTRL**-double-clicking on an object in a drawing will launch the **Fields** dialog.

**Keyboard Shortcuts**

Learn to use Windows keyboard shortcuts to choose items in dialogs and in panes. Manifold supports the usual Windows "power moves" to choose items in tables, dialogs and panes. If you are new to Windows, these are essential power moves similar to those used in many applications:

- **Click** Click on a record handle to select it.
- **CTRL-click** CTRL-click on a record handle to select it without de-selecting other records. **CTRL-click**-clicking on a selected record handle will de-select it without changing selecting of other records.
- **SHIFT-click** SHIFT-click on a record handle to select it and to also select all the records between it and another selected record. This is often used to select a series of records: click on the top record handle and then **SHIFT**-click on the bottom record handle and all the records in between will also be selected.
- **CTRL-A** Select all items
- **CTRL-I** Invert Selection.
- **CTRL-X** Cut selected items and copy to the Clipboard.
- **CTRL-C** Copy selected items to the Clipboard.
- **CTRL-V** Paste the contents of the Clipboard into the active window.
- **CTRL-N** New. Open a new project.
- **CTRL-O** Open. Open a previously created project.
- **CTRL-P** Print.
**CTRL-S** Save. Use this to save projects regularly.

**CTRL-Z** Undo the last command (if undoable).

**CTRL-Y** Redo the last undone command (if redoable).

**CTRL-*** (Asterisk on the numeric keypad) With focus on an image component’s window, zoom an image to the native zoom, with one image pixel equivalent to one screen pixel.

**CTRL- +** (Plus sign on the numeric keypad or in the regular keyboard) Zoom in.

**CTRL- -** (Minus sign on the numeric keypad or in the regular keyboard) Zoom out.

**CTRL- 0** Control Zero (in the regular keyboard) - Zoom to Fit.

**CTRL- *** Pressing **CTRL- *** (Asterisk) with the focus on the Project pane expands all components.

**CTRL- /** Pressing **CTRL- /** (slash) with the focus on the Project pane collapses all components.

Note that the asterisk, plus sign and minus sign on the numeric keyboard are not the ‘same’ characters from a shortcuts perspective as the equivalent characters on the main keyboard. These shortcuts work only from the numeric keypad.

When used in maps, Clipboard commands operate only on the active layer.

**Keyboard Accelerators**

Menu and button commands in Windows will often have a keyboard equivalent, which is a single character pressed together with the **ALT** key. For mnemonic purposes, the character chosen is usually one that occurs in the command caption, ideally the first character in the caption if it is not already in use for another command.

Keyboard accelerator characters are normally underlined to let us know they are available. If they are not underlined due to Windows system option settings, underlining may be turned on by pressing the **ALT** key once.

In the example above at right, we’ve pressed the **ALT** key once to turn on underlining. We can see that the **Accept** command button has an **ALT-e** accelerator and the **Look Up** command button has an **ALT-L** accelerator. Pressing **ALT-L** on the keyboard will have the same effect as clicking the **Look Up** button with the mouse.

Windows can be instructed to automatically show underlines for accelerator keys by checking the **Show extra keyboard help in programs** box in the **Keyboard** pane of the Accessibility Options applet in the Windows Control panel.

**Hand Positions for Keyboard and Mouse**

Intermediate and expert users will often use keyboard shortcuts together with mouse motions. One hand is on the keyboard and one hand on the mouse. Keyboard shortcuts are best arranged for right-handed users.
For editing operations, keep the right hand on the mouse and the left hand on the keyboard. For example, with images the mouse is used to select regions of pixels and to click on different windows or to choose layers. The left hand is used to make CTRL-X, CTRL-C and CTRL-V commands to cut, copy and paste to and from the Clipboard. Touch typists will often do a one-handed CTRL-C with the little finger on the CTRL key and the forefinger on the C or V. A CTRL-Z is immediately at hand if an editing operation needs to be undone.

Navigation also goes faster with this arrangement. Use the CTRL key to hot scroll by CTRL-right-clicking with the mouse into the active window to scroll it in the desired direction. When the mouse is in Zoom mode, the ALT key is used to reverse Zoom from Zoom in to Zoom out.

**Keyboard Shortcuts for Panes**

In addition to the standard Windows keyboard shortcuts, Manifold provides a set of keyboard shortcuts for opening/closing panes within the overall user interface.

<table>
<thead>
<tr>
<th>ALT-SHIFT-(letter)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>K</td>
<td>Call Stack</td>
</tr>
<tr>
<td>C</td>
<td>Control points.</td>
</tr>
<tr>
<td>E</td>
<td>Errors</td>
</tr>
<tr>
<td>G</td>
<td>GPS Console</td>
</tr>
<tr>
<td>H</td>
<td>History</td>
</tr>
<tr>
<td>I</td>
<td>Info</td>
</tr>
<tr>
<td>L</td>
<td>Layers</td>
</tr>
<tr>
<td>N</td>
<td>Notes</td>
</tr>
<tr>
<td>P</td>
<td>Project</td>
</tr>
<tr>
<td>R</td>
<td>Review</td>
</tr>
<tr>
<td>S</td>
<td>Selections</td>
</tr>
<tr>
<td>T</td>
<td>Tool Properties</td>
</tr>
<tr>
<td>A</td>
<td>Variables</td>
</tr>
<tr>
<td>B</td>
<td>ViewBots</td>
</tr>
<tr>
<td>V</td>
<td>Views</td>
</tr>
<tr>
<td>W</td>
<td>Watches</td>
</tr>
<tr>
<td>O</td>
<td>World</td>
</tr>
</tbody>
</table>
ALT-SHIFT keyboard shortcuts are very important for turning panes off and on when working with systems that have small screens. If your screen display area is limited in size it will be inconvenient to open panes and leave them open because they will cover up too much of the active window. ALT-SHIFT shortcuts will allow you to instantly pop open a pane when needed and to make it vanish when it is not needed.

These shortcuts are listed in the View - Panes menu next to the pane they control.

**Keyboard Shortcuts to Import Components**

<table>
<thead>
<tr>
<th>CTRL-SHIFT-(letter)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>Comments</td>
</tr>
<tr>
<td>D</td>
<td>Drawing</td>
</tr>
<tr>
<td>I</td>
<td>Image</td>
</tr>
<tr>
<td>Q</td>
<td>Query</td>
</tr>
<tr>
<td>S</td>
<td>Script</td>
</tr>
<tr>
<td>T</td>
<td>Table</td>
</tr>
<tr>
<td>U</td>
<td>Surface</td>
</tr>
</tbody>
</table>

Expert users should learn the keyboard shortcuts and use them. They will really boost efficiency.

**Keyboard Properties Shortcut**

Clicking ALT - ENTER will open up the View - Properties dialog for the active window.

**Keyboard Modifiers with Selection Tools**

- **ALT**
  When making selections in maps, ALT extends selections to all visible layers.

- **SHIFT**
  Shifts selection to "open" mode.
  
  In drawings select all objects any part of which is within the selection cursor. In images (using Select Touch) the SHIFT key will apply the touch select to all pixels like that one touched even if they are not contiguous.

  In images, a SHIFT modifier selects invisible pixels as well as regular pixels.

  In tables, a SHIFT click on a record selects the range of records between the record and other selected records.

- **CTRL**
  No matter what selection mode is actually set, forces use of Invert Selection mode for that operation. When used with Select Touch in drawings or tables, for example, a CTRL click is equivalent to toggling the selection status of an object or record.

  Holding down the CTRL key while using Select Touch with images will select only that pixel that is touched. This is normally used when zoomed far into the image to select / deselect individual pixels.

**ALT Key in Maps**
Selection operations in maps work only on the active layer. To extend the action of these operations to all visible layers, press the ALT key. For example, drawing a selection box in a map will only select objects in the active layer. Holding the ALT key while drawing a selection box will select objects from all visible layers. Pressing Delete deletes selected objects in the active layer only. Pressing ALT-Delete deletes selected objects in all visible layers. Think of ALT as a mnemonic for "All."

**Keyboard Modifiers with Smart Mouse Selection**

"Smart mouse" selection occurs when the mouse is not occupied with any command mode. It is a default selection methodology used to pick out an object for editing in drawings. An object chosen for editing is also called the primary selected object.

- **Click** Equivalent to Select Touch in Replace mode. Click on an object to select it. All other objects are deselected. Clicking into an empty part of the drawing deselects all objects (with confirmation dialog).

- **CTRL-ALT-Click** Select an object for editing. Click on an object to select it as the primary selected object. All other objects are deselected. Clicking into an empty part of the drawing deselects all objects.

- **CTRL** Invert the selection state of the object without changing the selection state of any other object. Equivalent to Invert mode.

- **SHIFT** Select the object if it is not yet selected. Does not change the selection state of other objects. Equivalent to Add mode.

**Keyboard Modifiers with Paint Bucket Tool**

- **ALT** When used with the Paint Bucket, the ALT key causes the tool to ignore the tolerance setting. This is used to flood the paint bucket throughout the selection.

- **SHIFT** Pour color into non-contiguous pixels.

- **CTRL** Pour color while preserving the intensity level. Uses the Paint Bucket to colorize.

**Keyboard Modifiers with the Tracker Tools**

- **TAB** Jump to nearest snap location, if Snap To is on.

- **SHIFT** Close tracker line and show computed ellipsoidal area.

- **CTRL** Toggle display between native drawing units and real measurement units (metric or English).

See the Tracker topic.

**Bookmark and Breakpoint Keyboard Shortcuts**

Bookmarks may be set within any text component (comments, queries or scripts) window. Breakpoints may be set within scripts.

**Bookmarks**
Introduction

F8  **Toggle** - Set a bookmark if one is not present. Clear a bookmark if one is already set for this line.

SHIFT-CTRL-F8  **Go to Previous** - Go to the previous bookmark.

CTRL-F8  **Go to Next** - Go to the next bookmark.

**Breakpoints**

F9  **Toggle** - Cycle the breakpoint at this line between "no breakpoint", "breakpoint set" and "disabled breakpoint."

SHIFT-CTRL-F9  **Go to Previous** - Go to the previous breakpoint (includes both disabled and enabled breakpoints).

CTRL-F9  **Go to Next** - Go to the next breakpoint (includes both disabled and enabled breakpoints).

**Keyboard Shortcuts for Editing Scripts and Queries**

Comments, query and script windows support Ctrl-[] and Ctrl-Shift-[] key combinations.

Ctrl-[] Check the character near the cursor and, if it is a bracket character, jump to the matching bracket.

Ctrl-Shift-[] Check the character near the cursor and, if it is a bracket character select the text between the bracket and the matching bracket.

Recognized bracket combinations are { and }, [ and ], ' and ', " and " and # and # (used to delimit dates in queries).

**Keyboard Modifiers for Selection in Tables**

(mouse click) Click on any record handle to select the record in accordance with selection mode currently set (Replace, Add, Subtract, Invert or Intersect). Click on a cell to move the focus to that cell.

SHIFT SHIFT-click on a record handle to apply the selection to a range of records.

CTRL Equivalent to clicking on a record handle when clicking into a cell. CTRL-click on a cell selects the record in accordance with the selection mode currently set. CTRL-SHIFT-click on a cell selects the range of records.

**Keyboard Zoom Commands during Selection and Editing**

When making selections or editing objects (such as inserting areas using the Autocomplete feature) we will often want to zoom in or out in the middle of a selection or editing command. Use the + and - keys on the keyboard to do so. Usually the + and - keys in the numeric keypad are used.

+  "Plus" key: Zoom in at the current tool position.

-  "Minus" key: Zoom out at the current tool position.

The current tool position is the position of the mouse at the time the + or - key is pressed. When editing, using these keys allows us to easily zoom into a desired location, make a few precise clicks and then zoom back out and proceed at the usual scale.
Keyboard Shortcuts for Toolbars

When working with a toolbar like the transform toolbar it is often convenient to work directly from the keyboard and to use keyboard shortcuts to move around between combo boxes, etc.

**TAB**  Jump to next control (combo box or button or other control).

**SHIFT-TAB**  Jump to previous control (combo box or button or other control).

**ENTER**  Clicks the nearest button next to the current control. For example, pressing **Enter** when in the target combo box in the transform toolbar is the same as clicking the nearby **Apply** button.

**SPACE**  Click the button that has the focus. No effect if no button has the focus.

**(arrow keys)**  The **left** and **up** arrows move the choice up in a menu while the **right** and **down** arrows move the choice down in a menu.

**(letter key)**  In a menu, jump to the first item beginning with that letter, if it exists. For example, in the transform toolbar’s operator box pressing the **m** key will jump us to the **Maximum** operator at the head of the list of operators beginning with an “**M**” (case insensitive).

**Home**  Jump to the top of a list.

**End**  Jump to the bottom of a list.

**Page Up**  Jump one “page” upward in a list.

**Page Down**  Jump one “page” downward in a list.

The letter key, **Home**, **End**, **Page Up** and **Page Down** shortcuts are particularly handy when finding commands in lengthy combo box menus, such as the list of operators in the transform toolbar’s operator box.

Keyboard Navigation in Terrain Windows

In addition to the keyboard arrows, Manifold uses keyboard navigation shortcuts to control the 3D view seen in the Terrain window.

**W**  Move forward.

**S**  Move backward.

**Q**  Strafe Up. (Move vertically up)

**E**  Strafe Down. (Move vertically down)

**A**  Strafe Left. (Move horizontally left)

**D**  Strafe Right. (Move horizontally right)

**SHIFT**  Pressing the **SHIFT** key with any of the above key commands will increase the effect of that command.

**Up / Down Arrow**  Tilt view up / down to +60 or -60 degrees from horizontal.

**Left / Right Arrow**  Rotate view left / right.

  +  Increase field of view (up to 130 degrees).

  -  Decrease field of view (down to 30 degrees).
* Reset field of view to 90 degrees

/ Move camera position to center of terrain.

The keyboard shortcuts are designed to allow "two handed" fast keyboard navigation as favored by experts in various computer tactical games. *Strafe* is gamer-speak for moving sideways or up/down without rotation. A *Q* move is thus the same as moving straight up, as in a helicopter.

To fly upward from the terrain surface, make sure that *Terrain - Snap to Surface* is unchecked. Next, either use *Q* to fly directly upwards or use the keyboard up arrow to rotate the view upwards and then use *W* to move forward. The *Q* move lifts us up as if in an elevator or helicopter moving straight up. The up arrow followed by a *W* is like angling an airplane upwards and then flying forward and up the angle.

**Keyboard Shortcuts for Z Scale in Terrain Windows**

With the focus on the terrain window we can use the following keyboard shortcuts to modify *Z scale*.

<table>
<thead>
<tr>
<th>Shortcut</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page Up</td>
<td>Increase Z scale by .01</td>
</tr>
<tr>
<td>SHIFT-Page Up</td>
<td>Increase Z scale by .10</td>
</tr>
<tr>
<td>Home</td>
<td>Set Z scale to 1.0</td>
</tr>
<tr>
<td>Page Down</td>
<td>Decrease Z scale by .01</td>
</tr>
<tr>
<td>SHIFT-Page Down</td>
<td>Decrease Z scale by .10</td>
</tr>
</tbody>
</table>

**Launching Help from Windows**

Manifold Help can always be launched directly from Windows. Press the Windows *Start* button in the Windows taskbar, and then choose *All Programs, Manifold System* and then *Manifold System Help*. It is often convenient to launch Help directly from Windows. Some Manifold users will launch Help in this way and keep that Help session available on their Windows taskbar for easy reference.

**Windows Styles**

Illustrations in this manual use a variety of Windows styles in an attempt to appeal to a diversity of Manifold users, many of whom continue to use older Windows releases.

Many screen shots are based on Windows 2000 and the "Classic" style in Windows XP. Since many users still use Windows 2000 or (through force of habit) use classic style in Windows XP the intent is to make the illustrations seem more familiar to as many users as possible.

When installed in Windows XP Manifold will automatically configure itself to use Windows XP style for controls, as seen below in a screen shot of the column formatting dialog for table windows.
Buttons in Manifold will adjust their appearance according to the current Windows theme in use and so may differ slightly from illustrations depending on the theme specified by the user.

With Manifold releases beginning with 7.00 in 2006, topics covering newly-introduced features began using native Windows XP style in screenshots. The expectation is that as Manifold evolves the number of screenshots using more modern Windows styles will evolve along with the number of users of such styles.
Introduction

Getting Started

Very Important: Before starting Manifold, go to the Windows Control Panel Display dialog Effects tab and make sure that the Show window contents while dragging check box is not checked. In Windows XP this option is found in the Control Panel's Appearances and Themes - Display choice under the Appearance tab by pressing the Effects button. Checking this box will greatly slow down the system when displaying complex maps and images because it forces a refresh of the window contents with each change in mouse position while dragging the window.

Projects

Manifold's main documents are projects, which use Manifold .map project file format (.map = manifold project). Within Manifold the standard Windows menu items File - New, File - Open and File - Save commands will create, open and save .map project files.

Projects contain a portfolio of components. Components are images, drawings, tables, labels, charts, maps and other items.

Creating a New Project

1. Create a new, blank project with File - New. This opens the project pane.
2. Using File - Import, import any drawings, images, tables and other components desired.
3. To add blank components, use the File - Create menu to add new blank drawings or other components.
3. Use File - Create - Map to create maps that show the components in the project.

By default, when Manifold System starts up it will open with a new, blank project. To open an existing project, simply use File - Open to open the new project.

The Project pane

The project pane shows all components we have in our project. Open/close the project pane with SHIFT-ALT-P or with View - Panes - Project.
In the default hierarchical view used for the project pane, components that are linked to other components will be shown indented below their parent component. Drawings, for example, are always linked to a table that contains the data fields for the objects in the drawing. Images that are palette images will have a palette linked to the image. Surfaces will have a terrain component created by default for the surface and may have one or more subsidiary terrain components.

Names of components are marked with an asterisk * if they have been added to the project or changed since the last Save command.

Manifold works with only one project at a time. To work with different projects (say, for example, to copy and paste data between projects) we can launch two Manifold programs and switch between them.

**Images**

Images are photographs or scanned data composed of pixels and normally imported from "raster" file formats like .bmp, .gif, .jpg and similar. Most people are familiar with pixel image editors such as Microsoft Paint, Microsoft Photo Editor or Adobe PhotoShop.

Multichannel data sets will be imported as a series of images, one for each channel. Images can also be created from surfaces and other components. See the Tools - Make Image topic.

**Drawings**

Drawings consist of points, lines and areas and are normally imported from "vector" file formats like .dxf, .shp, .mid/mif, .mfd/mdb, DLG, SDTS and other formats. What many older GIS systems call "maps" are called drawings in Manifold. Many people are familiar with vector editors such as classic CAD editors like AutoCAD, or from vector style graphics editors like Adobe Illustrator. See the Import and Export topic for more information on specific import dialogs.
Unlike simple CAD or vector graphics editors, Manifold can attach database information to objects in drawings. Every Manifold drawing has a table linked to it. Each row in the table is a record that corresponds to an object (point, line or area) in the drawing. Each column in the table is a data field. Tables linked to drawings have at least one field, the object ID field that gives the unique, numeric identification number for each object in the drawing.

**Labels**

Labels are components that contain text annotations that can be manually created or automatically created from the information in a drawing's table.

Manifold has a vast array of label capabilities to allow us to create labels using different fonts, colors, styles and other characteristics. Virtually every characteristic of a label can also be controlled automatically based upon information in a parent drawing. For example, labels can be placed on lines so they follow the shape of the line as seen in the above illustration.

Automatic creation and control of label characteristics is very important because many times the projects we create in Manifold will be seen through views that are automatically created. For example, if we create a map with many layers in Manifold and that map is displayed to people through a Manifold Internet Map Server web site, each view seen in that web site will be automatically created based upon however that particular user zooms or pans into the map.

In such cases, what is seen in the map view, be it the lines of drawings or the positions of labels, will be automatically created by Manifold. If we learn about Manifold facilities for the automatic control of labels we can create a map that will look good and serve its purpose regardless of how it is viewed.

**Surfaces**

Surfaces are like images in that they consist of pixels. Surfaces are used to represent continuously varying data such as terrain elevation data and are imported from a variety of formats such as DEM, SDTS and similar.

Manifold shows surfaces by default in 2D displays like the one above, where the surface is seen from overhead with shading to make the ups and downs of the surface more evident. By default surfaces are shown using grayscale coloring, but Manifold makes it very easy to color the surface by its height such as in the illustration above.

**Terrains**

Terrains are 3D views of surfaces. Whenever Manifold imports or creates a surface it will automatically create a terrain for that surface as well.
By default a terrain is a ground-level view looking North from the approximate center of the surface, with the 3D terrain uncolored. It's easy to color the terrain as we like and to "fly" about the terrain to whatever view we like.

Maps

A map displays one or more drawings, images, surfaces or labels stacked up as layers. The "white space" in drawings is normally empty space that is transparent so objects in lower layers can be seen. Images and surfaces do not normally have any "white space" (although some images or surfaces may have regions of transparent pixels) so these are normally used as the lowest layer in a map or shown with partial transparency so that layers below them can be seen. Layers in maps may be made partially transparent using an Opacity setting.

Maps can show drawings and images in any projection desired. Even if a drawing was created in one geographic projection, a map using that drawing can show it in a different projection. Different maps using the same drawings or images can show those same drawings or images in different projections at the same time.

We can create as many maps as we like within a particular project, and those maps can mix and match components from the project however we wish. The maps do not duplicate the components they contain as layers. Maps are just a way of viewing and organizing components in layers. Maps can contain layers that are drawings, images, surfaces or labels.

Creating a New Map
We can create a new map using whatever source materials we like:

1. Create a new project using File - New. This opens a project pane.
2. Using File - Import, import any drawings and images into the project the new map will use.
3. Create a new map in the project using File - Create - Map. We must specify the name of at least one drawing or image this map will use. In the Create Map dialog, check all the drawings or images from the project you would like to appear in this map.
4. Double click on the new map in the project pane to open it in a map window.
5. Drag and drop any additional drawings and images from the project pane into the map window. This adds them to the map. To add layers to the map we can also right click on any existing layer tab and choose Add - Layers from the context menu.
6. If images are not already georeferenced, they must first be georegistered so that they appear in the proper geographic location at the correct size.
7. If desired, change the projection used by the map.

When a map is created it is always based on at least one drawing, image, labels component or surface. When a map is first created in a project it uses the projection of whatever component(s) it uses. If it was created using only one component the projection of that component will be used for the map. If a map is created using multiple components at the same time, then the map is created using the projection of the largest image or surface in the map. If there is no image or surface a random choice of one of the native projections of one of the constituent components will be used.

If we would like the map to use a specific projection that's used by one of the components it shows we can easily specify the projection to be used by right-clicking on any layer and choosing the Use Projection choice from the context menu. The map will automatically switch to using the native projection used by that layer.

Importing Drawings, Images and other Components

Use File - Import to import a new drawing, surface, table or image into a project. Importing a drawing will automatically create a table that is linked to that drawing. Import more than one item of the same type by using CTRL-click to choose more than one file when browsing in the Import dialog. We can also click on one file and then SHIFT-click on another to highlight all the files in between if we wish to choose several files for import.

Importing a component into the project copies the data into the project file. All of a project's components are normally kept in a single file, the project's .map (map = manifold project) file. Importing an image, for example, copies the data from the original image file into the .map file, making the .map file increase in size. Keeping all components within one file makes it easy to copy projects and to send them to other people.

Database tables may be either imported into the project or linked to an external database file or data source. Tables that are linked to an external source are not copied into the project even though they appear to be inside the project just like any other table. Data in externally linked tables is fetched "on the fly" from the external file or data source as it is needed.

We can also create linked drawings that are automatically created based on external database tables. We can also create linked images that are brought into a project from a variety of external sources.

Finally, when using Manifold Enterprise Edition a project can include any component that is linked into the project from a Manifold Enterprise server.

To save a component from the project into a non-Manifold format, use the File - Export command. For example, any image in Manifold can be saved into popular graphics formats such as .bmp or .jpg, and drawings can be exported into popular GIS formats.

Creating New Components

The File - Create menu will insert a "blank" component. It is often used to create blank "layers" for incorporation into a map or to receive objects that were copied from another drawing and are now being pasted into their own drawing.
A nuance: drawings, images, surfaces and labels components within geographic systems must be created with an awareness of geographic location and scale to be used together in a sensible way. Manifold will try to do this automatically by using either reasonable defaults or by taking into consideration the location and scale of the component that is active when the new component is created.

If no component is open the **File - Create** dialog will create the new component with a default Orthographic projection centered at the 0,0 world latitude/longitude origin. If a map, image, drawing, labels or other component window is active when the new component is created, the new component will be created using whatever projection is used by that active window. This context-sensitive setting of default projection parameters makes it much easier to create new components using projection parameters that are hassle-free by default.

We can also create a new, blank layer in a map by right clicking on any existing layer tab and choosing **Add - New Drawing** or **New Image** or **New Labels**. This will create a new component of the desired type in the project and add it to the map as a layer just above the active layer. The new component will be created using the map's projection except in the case where a **New Labels** command is used to create bound labels from a drawing. In that case the new bound labels component will use the same projection as the drawing upon which it is based.

### Components

Components are items such as drawings or images that appear in the project pane and make of a Manifold project. The data for components is normally stored inside the .map project file; however, some components (images, drawings, tables and surfaces) can be linked components that are created based upon data sources, such as database tables or streamed data from image servers, from outside the project. In that case, the linked component's data is not stored within the project but is dynamically fetched from the source.

Note that linked components can also be created from local components stored inside the project. For example, we might create a linked drawing that is created based upon a query where the query acts on a local table inside the project.

### Components include:

**Drawings**

Often referred to as **vector** documents or vector "maps" in older GIS systems, drawings are used in CAD systems for blueprints or in vector GIS systems for mapping. Drawings are made up of **objects** that may be points, lines or areas based on specific coordinates. Every Manifold drawing is linked to a table. Every object in the drawing corresponds to one record in the table. Manifold can import drawings from a vast array of different GIS and CAD formats. Drawings that are automatically created from database tables are called **linked drawings** and are shown using an icon that includes a yellow "database" cylinder.

**Themes**

Themes are displays of a drawing shown using different formatting. Manipulating a theme is just the same as manipulating its parent drawing. Themes make it possible to show drawings colored or presented in different ways without having to duplicate the drawing.

**Images**

Images are composed of **pixels** arranged in orderly rows and are often referred to as **raster images** in older GIS systems. Most of us know images as the familiar pixel-based images we edit in Microsoft Paint, Adobe Photoshop or Microsoft Photo Editor. Images in Manifold can also be multi-channel raster data as well as visual images. Images are normally imported into a Manifold project. One special type of images, **compressed images**, may be linked into a Manifold project from external files or compressed image servers. **Linked images** may also be created from queries and database tables as well as from image servers such as TerraServer or OGC WMS servers, in which case they are indicated with an icon that incorporates a "database" cylinder.

**Tables**

Database tables may be included either within a project or linked from external database providers so that the
data they contain is fetched "on the fly" as it is needed. Tables linked from external sources are shown with an icon that includes a yellow "database" cylinder. Manifold includes an immense array of database capabilities that work with tables and which may be used for database manipulation of tables from almost any database management system.

Labels

Labels components show text labels that are either entered manually or are automatically created from data fields linked to drawings. Labels components may be created in a variety of ways, such as making a Copy of drawing objects and then Paste As a Labels component.

Maps

Maps show drawings, images and text labels as layers in a map window. An important function of maps is showing the data they contain in projected form, where the native coordinates of drawings are transformed into a desired geographic or other projection and where images are referenced into the desired coordinate system and shown in projected form. Map windows are the main user interfaces within Manifold. Maps are used not only for geographic presentation but also for working with non-geographic images and drawings in many layers.

Surfaces

Surfaces are raster data sets that contain data values, such as elevations, for each location. They are almost exactly analogous to images except that instead of a color value for each "pixel" they contain a data value. The default way to show a surface is as a shaded relief 2D image. Surfaces are called "grids" in some GIS packages. Surfaces may also be linked from queries or tables, in which case they are indicated with an icon that incorporates a "database" cylinder.

Terrains

Terrains are surfaces shown in three-dimensional views. Every terrain has a parent surface. Terrains are most often used to show 3D views of the surface of the Earth based on terrain elevation data; however, they may also be used to see abstract data such as population gradients, temperatures or other data as 3D surfaces. Each surface can have an unlimited number of terrain views created for it.

Profiles

Profiles are subsidiary components of surfaces that show a path over the surface. They are used to compute elevations, which show the cross-section of height along the path. Profiles are visible in a project only if the optional Surface Tools extension has been installed.

Elevations

Elevations are subsidiary components of profiles that show the cross-section of height through the surface along the path. Elevations are visible in a project only if the optional Surface Tools extension has been installed.

Charts

Charts allow database table visualization and data mining using a simple 2D charting style.

Palettes

Certain types of images compress data by saving a single number for each pixel that is an index into a lookup table of colors. A palette is the lookup table that specifies what colors to use for different pixel values. Manifold allows different palettes for use with images. These may be kept within the project as separate components. This is especially useful when using "false color" to view multispectral raster data images.

Layouts

Layouts allow the creation of different arrangements for printing the project or exporting a layout to a format such as .pdf or .ai. One might save several different layouts for different sizes and resolutions of printers, for example.
Queries

Queries are saved queries written in Manifold SQL. Manifold includes a complete SQL engine as extended with high-performance spatial, networking, logical and "fuzzy" predicates. Query windows are unusual in that they support multi-level Undo / Redo.

Scripts

Scripts are written in any Microsoft .NET language or any ActiveX scripting language such as Visual Basic Scripting Edition, JavaScript (both included by default with Manifold System) for which an ActiveX scripting engine is available on the system. Scripts may be used to extend the capabilities of Manifold System or to customize the appearance and function of existing capabilities. Script windows are unusual in that they support multi-level Undo / Redo.

Forms

Manifold includes the ability to create script-based forms or dialogs that provide various controls and call other scripts. Forms are often used to create simplified interfaces for inexperienced users.

Comments

Written notes and comments we wish to make. Comments components are also used whenever text reports are created within Manifold. For example, importing from formats such as SDTS will often create comments components that contain the metadata text embedded into the SDTS files. Comments windows are unusual in that they support multi-level Undo / Redo.

A Window for Everything

Double click on a component in the project pane to open it within its own viewing window. Alternately right click on it and chose Open. If a component is already open, double clicking on it in the project pane will bring that component window to the foreground. To open multiple copies of a component, right click on the component and choose Open in New Window.

Components will open in a window with characteristics appropriate for that type of component. Tables will open in table windows, surfaces will be seen in 2D shaded relief, terrains will be seen in 3D view windows and drawings, images and labels will be seen in their own types of windows.

Most Manifold windows will use progressive rendering, which enables other commands to function even while the window is still being painted. This is very useful with large, complex components that might take some time to render. For example, even before a large drawing finishes rendering we can begin zooming in to a region of interest. When a component is small enough to render very quickly, progressive rendering will not be noticed because the component will be rendered in a single pass.

Maps are seen in a special type of window that has layer tabs at the bottom. Maps are the main user interface in Manifold because maps show components like drawings, images, surfaces and labels components in layers. Maps show their contents in a stack of multiple layers and can apply a variety of effects such as opacity / transparency that work only with layers. Drawings and images are normally viewed through the map that includes them. Labels components are almost always used within maps.

If desired, we can double click directly on any drawing, image, surface or labels component and see it in its own window. There are two main differences between a map window and opening a component by itself in its own window:

- Maps can contain multiple drawings and/or images in layers, so map windows have layer tabs and other controls for working with layers. Each drawing window or image window works only with one drawing or image so it has no tabs or other controls for multiple layers.
- Maps can use projections that are different than the native projections of the drawings and images they contain. Map windows will re-compute coordinates on the fly so that all layers appear properly within the projection used by that map. A map window can work faster if the map and the components it contains all use the same projection.

We can open as many windows as we like, including multiple windows for the same map, drawing, image, table or other component. Each window may be operated independently to allow different views into the same
component. All windows viewing a given component will be updated to reflect any changes made to the component as they are made in any window for that component. Components will "remember" their previous settings, such as selections that were made, and show them in any window in which they appear.

3D Rendering Speed

Terrain windows require a video graphics card using a reasonably modern graphics processor and OpenGL acceleration in hardware for smooth response in their 3D display. The cost of modern graphics hardware, like a graphics card using a fast nVidia graphics processor, is very low so it pays to get a really fast card.

No Scroll Bars

To navigate within Manifold image, drawing or map windows use the Center button, the Grabber or hot scroll commands using a ALT-right click. ALT-right-clicking near the edge of the screen will scroll it in that direction. Hot scroll works faster than scroll bars and requires less accuracy with mouse clicks. It also saves screen space that is wasted when many windows are open at once and each has scroll bars.

See the Windows topic for more information on hot scroll.

The Startup View

By default, component windows will open up to show the view used when the component was last opened. If the last time we opened a map the window showed a zoomed in view and then we close the window, the next time we open that map the view will also be zoomed in. This behavior is controlled by the Save last used view in component option in the Tools - Options - User Interface dialog.

If desired, we can also designate a startup view to be used whenever that component is opened regardless of how it was last displayed. See the Views pane topic for more information.

Panes

Some dialogs are so frequently used within Manifold that they are implemented as panes. Panes may be left open "always on top" or docked into the main Manifold window. The Selections pane, for example, shows any saved selections for the current component. The Layers pane provides a extended list-style presentation of layers that is easier to use than layer tabs when a map contains many layers. Panes will automatically update their contents to apply to whatever window has the focus.

Use SHIFT - ALT keyboard shortcuts to turn panes on and off rapidly. Experienced Manifold users will almost always turn panes on and off with SHIFT-ALT commands (such as SHIFT-ALT-S to turn the Selections pane on and off) instead of using the View - Panes menu. See the Windows topic for more information on keyboard shortcuts.

Intelligent Menus and Toolbars

Manifold knows which window is active. Toolbars and menus will automatically adjust so that the commands they present are appropriate to that window. Commands that do not make sense (such as pixel selection when working in a drawing window) will automatically be disabled. Panes that are left open will automatically update their contents to show status for the active window.

System Activity Indicator

The System Activity indicator in the status bar at the lower right hand of the screen changes shape whenever Manifold is busy executing a command.

When the system is busy it will show a 3D "bump".

If desired, unchecking the Quiet system activity indicator in the Tools - Options User Interface page will change the indicator to use colors.
The default color for system idle status is green. This may be changed by changing the **System Activity Indicator Idle** color in the **Tools - Options - Colors** page.

The default color for system busy status is red. This may be changed by changing the **System Activity Indicator Busy** color in the **Tools - Options - Colors** page.

Some users prefer a less-intrusive 3D bump, while others prefer a dot that changes color. Some users may want a quiet indicator when the system is not active but a colored indicator when the system is busy. To achieve this effect, uncheck the **Quiet system activity indicator** option and to specify the same gray color as is used for Manifold window frames as the **Idle** color. The dot will then change from a "quiet" indicator to red when the system is busy.

If not in quiet system activity mode, the system activity indicator will show a busy color indication if the system is doing something in the background. For example, if the system is fetching tiles from an OGC WMS image server in background (see the Linked Images from OGC WMS Servers topic) it will show a busy indication.

**Sounds**

At times we may launch a very demanding command that will take some time to accomplish. Some jobs may be so computationally intensive that they may require hours or longer to run. It is often convenient to set such jobs running while we continue work on another computer; however, in such cases it is also convenient to have Manifold play a sound or otherwise alert us when the job is done.

The **Tools - Options - Sounds** dialog allows us to specify whether or not we want a sound to be played at the conclusion of a lengthy process, how long the process must be before a sound is played and, optionally, a **.wav** file from which a sound can be taken.

**The Status Bar**

The status bar at the bottom of the screen displays information on command status, location and other useful information. The leftmost region of the status bar is used to display brief descriptive phrases or captions relevant to the current command, somewhat like extended tool tips. Not all commands will show such descriptive phrases.

The rest of the status bar provides readouts that apply to the currently active window. Not all readouts will be active for all types of windows. For example, the **Location** and **Scale** readouts will not be active for table windows. The **Scale** readout will be active only for projected components and maps. Options specified in the status bar page of **Tools - Options** may be used to turn on and off the readouts in the status bar shown below.

<table>
<thead>
<tr>
<th>Orthographic</th>
<th>-108.8464 25.4957</th>
<th>&gt; 1:610000</th>
<th>Selected: 0</th>
<th>Snap</th>
<th>Replace</th>
<th>Zoom</th>
<th>⏯️</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Projection</strong></td>
<td><strong>Location</strong></td>
<td><strong>Scale</strong> or <strong>Size</strong></td>
<td><strong>Selection</strong></td>
<td><strong>Snap</strong></td>
<td><strong>Selection</strong></td>
<td><strong>Command</strong></td>
<td><strong>Mode</strong></td>
</tr>
</tbody>
</table>

- **Projection**: The name of the projection (coordinate system) in use.

- **Location**: The current location of the mouse cursor, usually in longitude and latitude. Can be in X, Y native coordinates for the projection. Locations are always given **Longitude** first and then **Latitude** in the traditional X, Y order, since longitudes are X, horizontal coordinate values and latitudes are Y, vertical values.

- **Scale or Size**: Enabled only if the component is projected. Show the current scale using the format chosen in **Tools - Options - Status Bar**. Options are:
  - **absolute scale** (:#:#) - Absolute scale in 1:xxx form.
  - **relative scale** - Relative scale as ratio between one centimeter or inch and the scaled number of meters/kilometers or feet/miles, as in 1 in : 364.84
miles

**horizontal extent** - Shows the horizontal size of the current window.

Choice of English or metric units is set in **Tools - Options - User Interface**.

**Selection**

The number of objects selected, if any in drawings, labels components and tables.

For surfaces, this status bar position will show the height of a surface layer at the current mouse cursor position. For map windows containing multiple surfaces, if a surface layer is active the status bar will report the height of the uppermost visible surface layer.

**Snap**

Shows “Snap” if a snap mode is active.

**Selection Mode**

Current selection mode: Replace, Add, Subtract, Invert or Intersect.

**Command**

Name of the current mouse mode command.

**System Activity**

Changes shape into a 3D “bump” whenever Manifold is busy executing a command.

The location readout may be changed in Tools - Options to show the location in different ways. Likewise, a **Tools - Options - Status bar** setting allows switching between different styles of showing scale.

The illustration above shows alternate choices for location and scale readouts. The location shows the longitude and latitude in degrees, minutes and seconds format. The scale readout shows the width of the current view in kilometers.

**Select objects within box given by two corners**

The leftmost portion of the status bar has multiple uses. It normally is used to show extended messages that accompany tool tips when the mouse cursor hovers over a command button. With commands like the Tracker the leftmost portion of the status bar will show the tracker readout.

**2135.9333 x 1996.6333 m**

For some commands, like selection commands, the leftmost portion of the status bar will be used to show the dimensions of the area selected in the units used for the current projection taking into account the local scale and units settings in the **Edit - Assign Projection** dialog for that component. If the component is georegistered the sizes shown will be true dimensions.

"On center" commands that start with a central point and then draw a circle, box or ellipse will report both the dimension and the central location used by the command.

**8824.8762 x 8824.8762 m @ -109.2391 25.6456**
For example, in the Select Circle command seen above the circle fits into a box 8824 meters wide and high and it is centered at approximately -109 longitude and 25 latitude.

**Projections**

Manifold can work with data saved in both projected and unprojected coordinate systems from all standard GIS formats. Unprojected maps are usually easy to import automatically no matter how geographically unaware the GIS format being imported. GIS data saved as projected maps might require user intervention in some cases if the format does not save the projection parameters necessary to use the data.

Any image or drawing or map can be re-projected using a virtually unlimited choice of standard projections offered by Manifold. See the Projections and Imported Components and Projections topics for details.

**Georegistering Images and CAD Documents**

Many images and CAD documents that we would like to use in GIS projects will come to us stored in file formats that, unfortunately, fail to save the projection and georegistration information that is required to easily use such images and CAD documents.

Perhaps, for example, we have an aerial photograph of a school campus that we would like to use as a backdrop for a nice map of the campus. Or, perhaps we have architectural plans of a new factory that we would like to show within a site map. Manifold has tools to enable us to use such data even if it originates in formats that are geographically unaware.

The process of adding these non-geographic documents to a geographic map and moving, re-projecting or resizing them to the correct position is called georegistration and is covered in the Georegistration topic. Although the process is simple, it must be learned and correctly applied if images are imported from formats that do not save georegistration information.

Manifold will automatically apply a projection to all images and CAD drawings (by default, the Orthographic projection) imported from geographically unaware formats. This is the right starting point for most non-geographic images and drawings that are to be used in maps. If desired, another projection may be specified if it is a better fit to the data, or images or drawings may be georegistered into a more precise match to any given component.

**Database Operations**

Manifold includes a very rich set of capabilities for working with databases either as separate tables or as tables linked to objects in maps. Although a major use of Manifold is to work with tables linked to drawings within maps, the database capabilities of Manifold are so strong that many users will use Manifold to view, analyze and manipulate database tables that have nothing to do with drawings. Quite a few people use Manifold as a personal information manager, since tables are easy to create in Manifold.

Database operations are accomplished using a combination of table windows and the table commands used in table windows, the Database Console, SQL queries, transform toolbar, Active Columns within tables, and ViewBots applied to tables. Some of the menu commands used with tables can be quite large subsystems, for example, as in the case of the Decision Support System. See the Rank Columns / Decision Support System topic for an example using Decision Support.

A useful Manifold subsystem used with databases is the Chart system. Charts are created from tables and then configured using a variety of presentation options and data grouping options.

**Programming**

Manifold may be programmed in several ways:

- "Ad hoc" programming using Active Columns in tables or ViewBots.
- Using SQL queries to perform analyses, alter table structure and otherwise make updates.
- Writing scripts using Visual Basic scripting, Javascript or other ActiveX scripting languages.
• Calling Manifold System from external programs.

There is no additional programming language or toolkit to purchase in order to program Manifold. Every system includes the ability to use Visual Basic scripting or JavaScript to write sophisticated, forms-based programs using standard Windows controls. If you do have a programming environment such as Visual Studio, you can use that as well without the need to purchase any other options.

See the Programming Manifold topic for more information on programming.

**Printing**

There are two ways of printing within Manifold:

- The contents of any component window can be immediately printed using default settings and simple options.
- For more sophisticated control over print layout, a **Layout** is created and opened. Components that are to appear in the layout are dragged and dropped into the layout and then resized, moved and adjusted as desired. We can then print the layout.

The most sophisticated print layouts will normally feature a main compositional element that is a map. This map will consist of many drawing, image and labels components to create the visual effect desired. Because maps provide very rich editing capabilities they are the natural "console" within which to work when creating desired effects.

Experienced Manifold users will therefore often put most of their compositional efforts into arranging the main map as desired. They will then use that map as the main element in a layout and will add other components to the layout rather sparingly. It is often a matter of taste and convenience whether a particular compositional element is placed within a map or is dropped into the layout as a separate element.

Suppose for example we have a map of wetlands that is to be printed and we wish to have a title block that includes a main text title hovering over a corporate logo. We could create the entire ensemble with both the wetlands drawings and the text labels and image used as the title block as a set of layers in one map, as illustrated at the end of the Layers topic. This map could then be dropped into a print layout that consists of just that one component.

Alternately, we could create a map showing the wetlands and then create another map that just consisted of a few labels and image layers that made up the title block. We could then drop the wetlands map into a layout and also drag and drop the title block into the layout. We would then print the layout that now is made up of two components.

The former approach allows creating everything within the same map window so that all layers are immediately at hand to allow adjustment of any visual aspect of the map. The second approach allows us to create a standard title block that could be recycled into other print layouts. Most production print jobs will include a combination of the two methods.

**Typical Workflow**

The workflow in mapping projects aimed at visual presentation consists of a few basic steps:

- Decide which drawings are to be used.
- Import those drawings into the project.
- Format those drawings for a pleasing and efficient basic appearance.
- Arrange the drawings as layers in a map.
- Project the map into an appropriate projection, if desired, and project the layers to match the map's projection (for speed).
- Create any new drawing layers in the map as necessary.
- Edit drawings to remove or adjust objects or to move them between map layers.
- Working in the map window, refine the formatting of the drawings for the exact visual effect desired.
- Improve the presentation with additional layers that contain text labels or images.
• Create a print layout and drag the map into the layout.
• Edit the print layout, adding any elements, resizing elements and placing elements as desired.
• Print the layout or export it to a .pdf or other file format such as .ai for further processing in other programs.
• If desired, publish the map to the web using the Manifold Internet Map Map Server.

Each of the above steps may involve nuances. For example, we will likely open and review various drawings before we decide which will be used in our project. If we don’t have necessary drawings on hand we will have to locate them or create them.

Importing drawings from other GIS or CAD formats may be very easy, or it may involve some detective work to uncover what projections or other parameters were used to create them and manual adjustment of drawing properties.

It’s often the case that drawings we use in a map will require editing. For example, perhaps we want to create a map of France and Germany but the only drawings we have available show all of Europe. We can copy only the objects for France and Germany and paste them into a new drawing that we will use.

Or, perhaps we are using a large, complex data set like TIGER/Line where part of the art of using the drawings is getting rid of very many unwanted items such as lines that show various grids or special zones in which we have no interest.

Editing will often make use of numerous Manifold tools and will require more advanced skills. For example, we might want to create a map of a single county that shows streams and rivers in the county, but perhaps the drawing of streams that we have shows streams as continuous lines that extend past the boundaries of individual counties. In that case, we’ll have to learn enough about Manifold editing tools to “cookie cut” the streams with county outlines so that we can delete those parts of the streams lines that extend past the county boundary.

Formatting will often involve thematic formatting, which uses data fields from the table linked to a drawing to automatically color or otherwise format the drawing. Depending on the visual effects desired, we may spend a substantial amount of time making adjustments in formats, editing drawings, creating new layers (drawings) and moving objects between layers to achieve exactly the effect desired.

When Manifold is used for serious analytic work we will often work with tables and the data they contain. In addition to the above workflow steps we will usually open tables associated with drawings and make selections, create ViewBots or perform other analyses. For example, we might use drawings as a visual means of selecting data in tables and then export that selected data into external database tables or Excel .xls spreadsheet tables.

When maps are used for visual exploration of data we may add facilities such as Zoom Ranges to improve navigation or to better present the data.

.map File Compression

By default, .map files are automatically compressed when saved to disk and automatically decompressed when read into Manifold. The compression ratio achievable varies depending on the types of data (components) that are in the .map file. A typical mix of components and data will usually result in a factor of five or seven compression compared to the size of a .map without compression. .map files consisting of large, mostly empty images and surfaces can result in compression ratios of up to 1 to 100.

Compressed .map files will automatically be decompressed when loaded into Manifold. Uncompressed .map files will be read directly into Manifold.

Compression does require additional time when opening and saving .map files. Users blessed with very large disk storage space may wish to turn off compression (in Tools - Options ) to enjoy faster loading and saving of .map files. In addition, compression requires additional working space on disk because compressed .map files are first decompressed to a temporary file and then loaded into Manifold. One must therefore have enough free space on disk to have a copy of the fully decompressed temporary file on hand. Considering that a 20 megabyte compressed .map file can decompress into a 100 megabyte temporary file, one should always have ample free space on disk for temporary files.

Manifold’s compression algorithms are lossless and do not cause any changes in the data. They are similar to the compression algorithms used in PC and UNIX “zip” style programs and result in similar compression ratios. When saving a compressed .map file to disk there is no gain to be had in further compressing it with a “zip”
program. One can therefore send compressed .map files through the Internet without feeling a need to further compress them with WinZip or similar utility to save space and transmission time.

**Setting Background Color**

The default background color for all components is set in the Tools - Options dialog. Each individual component can have its background color set by opening the component in a window and choosing View - Properties.

**Notes**

The Tools - Options dialog contains many options to control Manifold configuration. For example, if desired, we can use Tools - Options to tell Manifold not to launch with a blank project.

Some options are very important. For example, the Import empty columns in tables option is not checked by default. Columns in tables that contain no values will not be created when a table is imported. To preserve identical table structure even if the table does not contain any data in some columns, check this option.
Creating a Project

Manifold documents are projects. A project is a Manifold .map (for “manifold project”) format file that can contain many different types of information. There is no maximum limit to the size of a Manifold project.

Projects may contain various components such as maps, drawings, images, surfaces, labels, tables, palettes, terrains, themes, charts, print layouts, SQL queries, scripts and script forms. The Project pane shows all components in the current project. By default, the project pane will show a hierarchical view where components linked to other components are shown indented below their parent component.

File - New and File - Open are used to create new projects or to open existing projects. Creating a project or opening an existing project automatically opens the project pane. The main component is a map, which shows drawings and images as layers within the map.

Clicking a component in the project pane highlights it. CTRL-clicking a component toggles its selection state, thus allowing highlighting of more than one component, which is very useful if we would like to delete several components at once.

Creating a New Project

1. Create a new, blank project using File - New. This opens the project pane.
2. Using File - Import, import any drawings, images, tables and other components you plan to use.

3. To add blank components, use the File - Create menu to add new blank drawings or other components.

3. Use File - Create - Map to create maps that show the components in the project.

By default, when Manifold System starts up it will open with a new, blank project. To open an existing project, simply use File - Open to open the new project.

Creating a New Map

We can create a new map using whatever source materials we like:

1. Create a new project using File - New. This opens a project pane.

2. Using File - Import, import any drawings and images into the project the new map will use.

3. Create a new map in the project using File - Create - Map. We must specify the name of at least one drawing or image this map will use. In the Create Map dialog, check all the drawings or images from the project you would like to appear in this map.

4. Double click on the new map in the project pane to open it in a map window.

5. Drag and drop any additional drawings and images from the project pane into the map window. This adds them to the map. To add layers to the map we can also right click on any existing layer tab and choose Add - Layers from the context menu.

6. If images are not already georeferenced, they must first be georegistered so that they appear in the proper geographic location at the correct size.

7. If desired, change the projection used by the map.

When a map is created it is always based on at least one drawing, image, surface, theme, or labels component. When a map is first created in a project it uses the projection of whatever component(s) it uses. If it was created using only one component the projection of that component will be used for the map. If a map is created using multiple components at the same time, then the map is created using the projection of the largest image or surface in the map. If there is no image or surface a random choice of one of the native projections of one of the constituent components will be used.

If we would like the map to use a specific projection that's used by one of the components it shows we can easily specify the projection to be used by right-clicking on any layer and choosing the Use Projection choice from the context menu. The map will automatically switch to using the native projection used by that layer.

When importing a component into a project, the entire contents of that component are stored in a special section of the project file. For example, when importing a 10MB .bmp image into a project, the project .map file will increase in size by 10MB. While this may seem wasteful of disk space, given the very large sizes and low costs of modern hard disks it makes more sense to keep all project components organized within a single file. This makes it easy to backup files, to organize files well for version control and to know that no necessary information is forgotten when a project file is sent to someone else.

There are two exceptions in which data for components is stored outside the Manifold project file:

- Some components (images, drawings, tables and surfaces) can be linked components that are created based upon data sources, such as database tables or streamed data from image servers, from outside the project. In that case, the linked component’s data is not stored within the project but is dynamically fetched from the source.
- Enterprise Edition allows keeping all project components saved in a remote Enterprise server.

Beginning users are cautioned, however, that using linked components or Enterprise Edition requires slightly more attention than simply keeping all components in one project file as is done by default. If a project contains all components it is a simple matter to send the project to someone else or to move the project to a different location by simply sending or moving the .map file involved. If external database tables or other sources are linked into the project more thought and attention are required to share the project with a colleague.

Maps are Free

Creating a map in the project adds very little to the size of the .map project file. A map component contains very little data, since the contents of the map are taken from the drawing and image components it contains. The
actual map component consists of pointers to the relevant drawing and image components as well as a very small amount of housekeeping information (layer ordering, projection parameters, etc). Therefore, making copies of maps or making lots of maps in a project is an efficient way of creating and saving different views into the same data. There is virtually zero storage cost to adding another map to a project.

.map Files and Compression

By default the .map format used to save Manifold projects automatically compresses the data contained in the .map file using techniques similar to those used by popular "zip" compression utilities. Therefore, usually there is no need to "zip" a .map file with utilities such as WinZip to save disk space or to speed up transfers over Internet - the .map file is already compressed.

Compression takes some time when saving a .map file and when opening a .map file to decompress the contents for use within Manifold. For large files the time required for compression and decompression may be significant. If speed in loading and saving is more important than saving space on hard disk we can turn off automatic .map file compression by unchecking the Compress .map files to save space option in Tools - Options. Except for speed in loading or saving there is no difference in system performance if .map file compression is turned on or off.

Processors are getting faster much more rapidly than disk drives are getting faster. In fact, some processors are already so fast that they can compress and decompress data faster than disk drives can read or write data. In such cases it is quicker to compress a .map file so that the relatively slow disk drive needs to spend less time reading or writing a smaller file. Therefore, with some fast systems it is wise to leave compression always turned on because that will not only save space on disk but it will also be faster. With slower systems, on the other hand, it could be that the compression option will result in more time required saving or loading .map files.

There is no indication that a .map file is compressed or uncompressed other than it takes more or less space. It is strongly recommended that if compression is turned off that the user establish operational practices to distinguish between compressed and uncompressed .map files. Newer processors are already so fast that it is almost always better to leave compression turned on all of the time.

The.map files on the Manifold downloads site were all saved with .map compression turned on in order to place more files onto the CD. When opened for use the resulting projects may be considerably larger than might be expected from the size of the .map file on CD.

ODBC and .map Files

Manifold includes an ODBC driver. This driver allows other applications to work with data in tables and queries in a Manifold .map project file. See the Using the Manifold ODBC Driver topic.

Tech Tips

Files ending in a .map extension may be used by other applications on your computer. When you install Manifold it will automatically tell Windows to open .map files with Manifold. If you would prefer to retain the assignation of files ending in .map to some other application, in Windows Explorer right click on a .map file and choose Open with and then Choose Program. Choose the program you wish to use to open .map files and check the box Always use this program to open these files and press OK.

We can rename project components by right clicking on them and choosing Rename in the context menu, which gives us the option to automatically adjust (rename) the names of any dependent components. An even faster way of doing this is to highlight a project component, pressing the F2 key (the usual Windows shortcut key to rename an item) and then after providing the new name finishing with a SHIFT-ENTER. Pressing ENTER alone after renaming an item renames only that item. Pressing SHIFT-ENTER after renaming an item renames that item as well as any dependent components.
Introduction

Components

Manifold System keeps the information it needs to keep track of items for a given project in a single file called a Manifold project file. Projects are the "documents" of Manifold, just like .doc files are the documents of Microsoft Word or .xls spreadsheets are the "documents" of Excel. A Manifold project contains all the different types of data with which we work, or, in the case of linked components, links to the source of the data used for that component.

The different types of data kept in a Manifold project are called components. Components are the drawings, images, surface, labels, comments, tables, scripts, queries, themes and other items in a Manifold project.

There are the following types of components in a Manifold project:

- **Local components** are components that exist entirely within the project. These are created in the project using Manifold dialogs, or they are imported into the project from some other file. For example, we might import a drawing from a shapefile. The data for local components resides entirely within the .map project file. If we import a 50 MB drawing from some external file and then save our .map project file, it will grow in size to accommodate the imported drawing.

- **Linked components** and **shared components** are components that are created on-the-fly from some other data source, usually some external database file, image server or from some other external source. The data for linked components usually resides outside of the .map project file. For example, if we link an image into a Manifold project from an external .ecw file that is 50 MB in size, when we save the .map project file it will not grow in size. The data for shared components always resides outside of the .map project file, but the project file may contain a cached copy of that data.

Manifold saves information to keep track of all of a project's components in the project's .map file. When a project is saved, any changes made to any component will be updated as well.

Because components are often imported from systems or formats where they exist as separate files (for example, .bmp or .gif images), it is tempting to think of components as being separate files. That's not the case. Even though components appear to be separate things they are all part of one Manifold project .map file. With the exception of linked components and Enterprise Edition projects they are all saved in the single .map file as well.

It is also tempting to think of components as somehow retaining the characteristics of their originating formats. For example, one might think that because a table was imported from a .dbf database format file it somehow is still limited to having restricted field names as if it were still in the original .dbf format. Or, perhaps we've imported a drawing from shapefile format and think the drawing must now be limited to the antique restrictions of that format.

That is not the case for components that are imported into a Manifold project. Once a component is imported into Manifold it is now in Manifold as a local component and has no tie to the original format or file from which it originated.
However, there may well be some restrictions on linked components depending on their origin. For example, linked images coming into the project from TerraServer will be read-only, and tables that are linked into a project from an external data source will be subject to the limitations of that data source. If we don't like such restrictions, we can always unlink the linked component to change it into a purely local Manifold component. We can then do with it what we please.

The Project Pane

Using File - Open to open a Manifold project or File - New to create a new project will automatically open the project pane. The project pane shows a list of all components in a project. Double click on a component in the project pane to open it within its own viewing window. Alternately right click on it and choose Open. If a component is already open, double clicking on it in the project pane will bring that component window to the foreground. To open multiple copies of a component, right click on the component and choose Open in New Window.

Some components in a project pane will be linked to other components and will be shown indented underneath those components in the project pane. For example, every drawing has a subsidiary table that will be shown indented underneath it and images using palettes will have their palette components linked to them. In the illustration above, the image SanFran is a palette image that uses the palette underneath it.

To rename a component, right click on it in the project and choose Rename. The Component Name dialog that pops up includes a checkbox to Adjust names of dependent components, which, if checked (the default), will automatically rename any subsidiary components. For example, if we rename a drawing called Mexico Drawing to Central Mexico Drawing, then the drawing's table that was originally called Mexico Table will automatically be renamed Central Mexico Table.

Components in the project pane may also be renamed by a “slow double-click” or by pressing the F2 key. Press Enter to commit the renaming operation without renaming dependent components, or press SHIFT-Enter to commit the renaming operation and to also adjust names of any dependent components as if the Adjust names of dependent components option in the Component Name dialog had been selected.

Types of Components

Components are items such as drawings or images that appear in the project pane and make of a Manifold project. The data for components is normally stored inside the .map project file; however, some components (images, drawings, tables and surfaces) can be linked components that are created based upon data sources, such as database tables or streamed data from image servers, from outside the project. In that case, the linked component's data is not stored within the project but is dynamically fetched from the source. Linked components are shown with a cute little database cylinder graphic added to the icon.
Introduction

Note that linked components can also be created from local components stored inside the project. For example, we might create a linked drawing that is created based upon a query where the query acts on a local table inside the project.

Components include:

**Drawings**

Often referred to as vector documents or vector “maps” in older GIS systems, drawings are used in CAD systems for blueprints or in vector GIS systems for mapping. Drawings are made up of objects that may be points, lines or areas based on specific coordinates. Every Manifold drawing is linked to a table. Every object in the drawing corresponds to one record in the table. Manifold can import drawings from a vast array of different GIS and CAD formats. Drawings that are automatically created from database tables are called linked drawings and are shown using an icon that includes a yellow “database” cylinder.

**Themes**

Themes are displays of a drawing shown using different formatting. Manipulating a theme is just the same as manipulating its parent drawing. Themes make it possible to show drawings colored or presented in different ways without having to duplicate the drawing.

**Images**

Images are composed of pixels arranged in orderly rows and are often referred to as raster images in older GIS systems. Most of us know images as the familiar pixel-based images we edit in Microsoft Paint, Adobe PhotoShop or Microsoft Photo Editor. Images in Manifold can also be multi-channel raster data as well as visual images. Images are normally imported into a Manifold project. One special type of images, compressed images, may be linked into a Manifold project from external files or compressed image servers. Linked images may also be created from queries and database tables as well as from image servers such as TerraServer or OGC WMS servers, in which case they are indicated with an icon that incorporates a "database" cylinder.

**Tables**

Database tables may be included either within a project or linked from external database providers so that the data they contain is fetched "on the fly" as it is needed. Tables linked from external sources are shown with an icon that includes a yellow "database" cylinder. Manifold includes an immense array of database capabilities that work with tables and which may be used for database manipulation of tables from almost any database management system.

**Labels**

Labels components show text labels that are either entered manually or are automatically created from data fields linked to drawings. Labels components may be created in a variety of ways, such as making a Copy of drawing objects and then Paste As a Labels component.

**Maps**

Maps show drawings, images and text labels as layers in a map window. An important function of maps is showing the data they contain in projected form, where the native coordinates of drawings are transformed into a desired geographic or other projection and where images are referenced into the desired coordinate system and shown in projected form. Map windows are the main user interfaces within Manifold. Maps are used not only for geographic presentation but also for working with non-geographic images and drawings in many layers.

**Surfaces**

Surfaces are raster data sets that contain data values, such as elevations, for each location. They are almost exactly analogous to images except that instead of a
The default way to show a surface is as a shaded relief 2D image. Surfaces are called "grids" in some GIS packages. Surfaces may also be linked from queries or tables, in which case they are indicated with an icon that incorporates a "database" cylinder.

**Terrains**

Terrains are surfaces shown in three-dimensional views. Every terrain has a parent surface. Terrains are most often used to show 3D views of the surface of the Earth based on terrain elevation data; however, they may also be used to see abstract data such as population gradients, temperatures or other data as 3D surfaces. Each surface can have an unlimited number of terrain views created for it.

**Profiles**

Profiles are subsidiary components of surfaces that show a path over the surface. They are used to compute elevations, which show the cross-section of height along the path. Profiles are visible in a project only if the optional Surface Tools extension has been installed.

**Elevations**

Elevations are subsidiary components of profiles that show the cross-section of height through the surface along the path. Elevations are visible in a project only if the optional Surface Tools extension has been installed.

**Charts**

Charts allow database table visualization and data mining using a simple 2D charting style.

**Palettes**

Certain types of images compress data by saving a single number for each pixel that is an index into a lookup table of colors. A palette is the lookup table that specifies what colors to use for different pixel values. Manifold allows different palettes for use with images. These may be kept within the project as separate components. This is especially useful when using "false color" to view multi-spectral raster data images.

**Layouts**

Layouts allow the creation of different arrangements for printing the project or exporting a layout to a format such as .pdf or .ai. One might save several different layouts for different sizes and resolutions of printers, for example.

**Queries**

Queries are saved queries written in Manifold SQL. Manifold includes a complete SQL engine as extended with high-performance spatial, networking, logical and "fuzzy" predicates. Query windows are unusual in that they support multi-level Undo / Redo.

**Scripts**

Scripts are written in any Microsoft .NET language or any ActiveX scripting language such as Visual Basic Scripting Edition, JavaScript (both included by default with Manifold System) for which an ActiveX scripting engine is available on the system. Scripts may be used to extend the capabilities of Manifold System or to customize the appearance and function of existing capabilities. Script windows are unusual in that they support multi-level Undo / Redo.

**Forms**

Manifold includes the ability to create script-based forms or dialogs that provide various controls and call other scripts. Forms are often used to create simplified interfaces for inexperienced users.

**Comments**

Written notes and comments we wish to make. Comments components are also used whenever text reports are created within Manifold. For example, importing from formats such as SDTS will often create comments components that contain the metadata text embedded into the SDTS files. Comments windows are
Introduction

unusual in that they support multi-level Undo / Redo.

**Importing Components into a Project**

To add a component from an existing file (such as a GIS format drawing or image file) choose File - Import to import a component from an existing file.

When importing or linking components we can usually import more than one item of the same type by using CTRL-click to choose more than one file when browsing in the Import dialog. We can also click on one file and then SHIFT-click on another to highlight all the files in between if we wish to choose several files to import.

Whenever we import a component such as a drawing, image or surface that can have a projection, if it has been imported from a format that does not store projection information when we first open it Manifold will remind us to verify the projection used for that component by popping open an information bar in the component window. Click on the bar to verify the projection using the Edit - Assign Projection dialog. If the projection shown in the Assign Projection dialog is what we think it should be, we press OK. Thereafter, we can open the component without Manifold nagging us to verify the projection.

*Note*: We can get rid of the info bar without launching the Assign Projection dialog by clicking the X sign at the right side of the info bar; however, Manifold will know that we've not yet verified the projection of this component and the info bar will appear again the next time the component is opened in a window.

If we do not want to verify projections of new components, once we acquire enough expertise not to want to verify projections anymore unless it is absolutely necessary, we can get rid of the info bar by changing the Prompt to verify projections of new components option in the Tools - Options dialog.

Conversely, if we would like Manifold to remind us to verify the projection of all new components, even those imported from formats that correctly store projection information, we can uncheck the Suppress prompt for non-default projections option in the Tools - Options dialog. See the discussion of info bar options in the Tools - Options dialog for details.

**Linking Components into a Project**

Choose File - Link to create linked components such as linked tables, linked drawings, linked images or linked surfaces within the project that take their data from external sources. Linked components will appear in the project like local components, but their data will come from an external source. Depending on the nature of that source, there may be limitations on those linked components. For example, linked images from image servers will be read-only and cannot be re-projected. Tables linked from external data sources using ADO.NET connection technology will be read-only. Drawings linked from geometry in database tables will normally be read/write and fully editable but usually cannot be re-projected.

**Creating New Components**

At any time we can create a new component in the project by choosing File - Create. Except for maps, the File - Create dialog takes the projection parameters for the new component from whatever window is active at the time the new component is created.

If no existing component is active, the new component will be created using the system default projection of Orthographic centered at the 0,0 world latitude/longitude origin. If a map window or other component window is active when the new component is created, the new component will be created using whatever projection is used by that active window. This context-sensitive setting of default projection parameters makes it much easier to create new components using projection parameters that are hassle-free by default.

When a map is created it is always based on at least one drawing, image, labels component or surface. When a map is first created in a project it uses the projection of whatever component(s) it uses. If it was created using only one component the projection of that component will be used for the map. If a map is created using multiple components at the same time, then the map is created using the projection of the largest image or surface in the map. If there is no image or surface a random choice of one of the native projections of one of the constituent components will be used.
If we would like the map to use a specific projection that's used by one of the components it shows we can easily specify the projection to be used by right-clicking on any layer and choosing the **Use Projection** choice from the context menu. The map will automatically switch to using the native projection used by that layer.

A very common way of adding components to a project is to create them from other components using **Copy** and **Paste As**. See the Copy and Paste As topic.

We can also create a new, blank component in a project by using the context menu for layer tabs in a map. Right click on any existing layer tab and choose **Add - New Drawing** or **New Image** or **New Labels**. This will create a new component of the desired type in the project and add it to the map as a layer just above the active layer. The new component will be created using the map's projection except in the case where a **New Labels** command is used to create bound labels from a drawing. In that case the new bound labels component will use the same projection as the drawing upon which it is based.

**Maximum Component Size**

No single drawing, table, image or surface can exceed 16 EB (exabytes) in size. Since one exabyte is $2^{60}$ bytes, about a trillion gigabytes, as a practical matter there is no size limit for these components. One will run out of disk space before the size limit is approached. There is no limit to the number of components that may be in a project.

Comments, queries and scripts are limited to approximately 1 GB. Labels are limited to 16 EB when initially created, but any changes made to a labels component after it is initially created cannot exceed approximately 1 GB.

Limits on comments, queries, scripts and labels component sizes are approximate because the precise limit depends upon details of the specific Windows system in use and the operation of that system. For example, in certain systems the 1 GB approximate limit on such components can expand to 2 GB or even 3 GB on machines run with a /3GB boot option. However, because such expanded uses depend on many complex interactions the stated limit size for comments, queries, scripts and changes in labels components is only 1 GB (a vast amount for such things).

Note that the 1 GB limit on "queries" is the size of the actual SQL query text, not on the results of a query. This is somewhat of an absurd "limit" since it is very rare that one writes an SQL query longer than a few hundred lines, let alone more than a megabyte in size.

**Opening or Running Components**

The different types of components that appear in the project pane can be opened for viewing or editing or, in the case of scripts, queries or forms, for execution (running) through mouse clicks, menu commands or toolbar buttons.

- **Double click** on a map, drawing, image, surface, terrain, table, comment, chart or palette to display it in a window.
- **Right click** and choose **Open in New Window** to open the component in a new window.
- **Click** on a form, script or query and press the Run button to execute it.
- **Double Click** on a form, script or query to edit it.
- **Right click** onto a component to pop open a context menu of choices for that component.

**Component Windows**

Double clicking on a component in the project pane will open that component in a window. Double clicking a map, drawing or image, for example, will launch the component within a window designed to display and manipulate that type of component. Component windows will automatically open with the correct set of toolbars or other controls appropriate to that type of component. Clicking on a form, script or query to highlight it and then pressing the Run button will execute that form, script or query. Queries will appear as tables when run. To edit the SQL text that generates a query table, open the query for editing by double-clicking it or by right clicking on it and choosing **Open**.

We can open multiple windows at once into the same component by right clicking and choosing **Open in New Window** from the pop-up menu. Each window can maintain its own independent view into that component. So, for example, we could open two windows to view the same image and zoom in on one window for a very close up
view while panning the other window to see a different part of the same image. Changes in any one window to a component (such as changing the formatting to change the colors it uses) will update all other windows for that component. If we change the background color for areas to bright green in a drawing all windows that show that drawing will show the areas in bright green background color.

Some components have other components that are associated with them. For example, every drawing is connected to a table for that drawing. If the drawing is open, we can open the associated table by choosing **Drawing - Open Table**. If the table is open in a window, we can open the related drawing by choosing **Table - Open Drawing**. These menu choices will be enabled only for components that are related to other components. For example, a table that is not connected to a drawing will not have an **Open Drawing** command available.

**Context Sensitive Controls**

Only one window at a time has the **focus** in Microsoft Windows. Clicking into a window or touching it in some way moves the focus to that window in the standard Microsoft way to make it the **active window**. When the focus is on a particular window its title bar changes to the highlighted color specified for active windows in the Display appearance settings specified in your Windows Control Panel. See the Windows topic for more on typical Microsoft Windows behavior.

Clicking on a component window moves the focus to that window. Simultaneously, the main toolbar controls will change context or form to support use of whatever component window has the focus. Manifold toolbars and toolbar buttons are designed to preserve commonality of function as much as is possible, so that when the focus changes from a drawing to an image window the selection buttons will retain their appearance. However, their function will change in subtle ways to support the slight differences in selection operations between objects in drawings and pixels in images.

When the focus moves to a new window, any panes open will automatically switch their contents to apply to that window.

**Saving / Exporting Components**

Once components (like drawings and images) are imported into Manifold they become part of the project **.map** file. Saving the project saves the components within the **.map** file. At times we would like to import images or drawings into Manifold and to later save them as independent drawing or image files.

To do this, we open the component in a window and then use the **File - Export** command to export it to the desired format. For example, we may import drawings in **.shp** format, manipulate them within Manifold, and then use **File - Export** to save them in **.shp** format for use with other, archaic software.

Note that not all formats can handle the full range of information possible in Manifold components. For example, tables in Manifold can contain field names that are fine for export to Access **.mdb** files but which are longer than allowed in the less-capable **.dbf** format used in **dBase** files. Such field names should be changed before attempting export to **.dbf** format.

Since **.shp** "shapefiles" are actually an ensemble of files that include the use of **.dbf** files for data attributes, this means that exporting a drawing (and, of course, its associated table) to **.shp** format means that field names in tables may need to be shortened.

By default, Manifold includes many more formats available for import than for export. Optional products from **manifold.net** will add additional formats for export. Visit the **manifold.net** web site for information on optional products.

**Tech Tip: Organize Projects with Folders**

The **create folder** button in the project pane toolbar creates folders within the project. Use folders to organize components within the project and to keep the project pane manageable.

This is especially important when importing drawings from complex file formats such as those used with VMAP and TIGER/Line, where a single import will create numerous drawings and maps. Imports will create components within the current folder. Before importing from such formats, create a new folder in the project to highlight it and then choose **File - Import**. Imported components will be created within the folder specified.
To move files between folders, drag components and drop them into a folder (drop them onto the folder icon itself and not the hierarchy beneath the folder). Folders may be created within other folders. If a folder is highlighted in the project pane and the create folder command is issued, the new folder will be created within the highlighted folder. Folders may be dragged and dropped into other folders. To move a folder out from within another folder, drag it and drop it into the "white space" of the project outside another folder.

.map Files and Compression

By default the .map format used to save Manifold projects automatically compresses the data contained in the .map file using techniques similar to those used by popular "zip" compression utilities. Therefore, usually there is no need to "zip" a .map file with utilities such as WinZip to save space or to speed up transfers over Internet - the .map file is already compressed.

Compression takes time when opening a .map file (to decompress the contents) and when saving a .map file. For very large files the time required for compression and decompression can be significant. If speed in loading and saving is more important than saving space on hard disk we can turn off automatic .map file compression in Tools - Options.

Use this option with care, since there is no indication that a .map file is uncompresses other than it takes more space. It is strongly recommended that if compression is turned off that the user establish operational practices to distinguish between compressed and uncompressed .map files. For example, one might never use compression on a particular machine, or one might keep all uncompressed .map files in a folder called "uncompressed."

Tech Tip: Save Frequently Used Components in .Map Files

We can keep collections of frequently used drawings or other components in .map files and then use File - Import - Component to import them from the .map file into the current project whenever desired.

Enterprise Edition

Manifold System Enterprise Edition also allows storage of components within centralized databases. With Professional Edition, every component that is in use in a Manifold project (with the exception of linked components such as linked tables, linked drawings, linked images or linked surfaces) is saved in a .map file that is used by one user at a time. If different users want to use the same drawing they must make a copy of it for use within their map files.

With Enterprise Edition, components may be saved within a centralized database, the Enterprise server, so that the same component can be shared by more than one user at a time. Enterprise users can also keep a single copy of a component within the Enterprise server and use it within many different projects. See the Enterprise Edition topic for more information. See the Data Storage Strategies for a discussion of desktop, shared enterprise and server storage models available within Manifold.

Choosing a Connection Technology

When linking components from an external data source we would like the link to be as fast and as capable as possible. Manifold supports a very wide array of connection technologies to allow informed users to choose the best possible connection method as they see fit. This can be somewhat daunting to new users who may not even know what ADO .NET is, let alone when to choose it over OLE DB.

In cases where general-purpose connection technologies such as ODBC or OLE DB are used to connect to data sources it is tempting to think of such technologies as "black boxes" that provide a seamless, two-way pipeline for data that provide the connection invisibly. In real life, however, there are very great differences between connection technologies and even within a particular connection technology there can be great differences between drivers that lead to significant performance differences or other limitations.

To cite just one difference between technologies, connections via ADO .NET can be 600 times faster to some data sources than the equivalent ODBC driver. To cite another difference between connection technologies, using Oracle's native Oracle Call Interface (OCI) provides better performance and allows using features not exposed through generic database interfaces, such as ODBC, OLE DB and ADO .NET. That is why Manifold uses OCI by default to connect to Oracle.
To cite a difference within the same technology, not all OLE DB drivers for the same data source are created equal: for example, the OLE DB driver provided by a DBMS vendor may be superior in terms of speed or capability to generic OLE DB drivers for that data source bundled within Windows.

The rule of thumb is to always use OCI for connections to Oracle. Manifold will do this by default whenever we choose Oracle data sources, and this default choice should not be replaced by something else unless the user is a serious expert on connections to data sources. For read/write connections to SQL Server, choose OLE DB using the SQL Native Client OLE DB driver.

For other data sources, use ADO .NET if a read-only connection is OK (such as when importing a table) and OLE DB otherwise. If you don't know whether a particular connection will be read-only or not or are worried whether any unexpected results you have (such as an inability to edit) result from using ADO .NET, use OLE DB always. If you think there may be a significant performance difference, visit the online forum for Manifold and discuss what you are doing with your fellow Manifold users and see what more experienced users advise.

Not all data sources may have either an ADO .NET or an OLE DB connection available, but virtually every data source anyone has ever heard of has had an ODBC driver written for it. Use ODBC when OLE DB or ADO .NET connections are not available.

Manifold has a lot of internal code devoted to working around various limitations that exist within some drivers for well-known data sources. For example, there is special code within Manifold to facilitate using OLE DB connections to upload binary data to MySQL even though it is not one of the "Big 3" DBMS packages officially supported by Manifold. To cite another example, establishing a database connection for a component linked from an ODBC data source will automatically degrade to read-only mode if connecting in read-write mode results in a connection that cannot be used to operate the component.

About the Nomenclature

Component was chosen as a deliberately neutral word to refer to the drawings, images and other items in a project. Manifold can work with so many different types of data that an existing GIS word like "layer" did not fit them all. For example, it doesn't make sense to talk about the "layers" in a project if some of the things in the project are scripts or comments or terrains, which can't appear as layers in a map. Also, some existing GIS words, like "theme" to take just one example, already mean something specific that doesn't necessarily work the same way in Manifold.

The word component is not intended to convey any special technical meaning other than indicating an item that is a part of a Manifold project. There is no fancy secondary meaning having to do with object-oriented programming or anything like that. It's just a generic, neutral word that ended up getting picked years ago.

If you are joining Manifold from some other GIS, please make an effort to use the word component and not some other word from that other GIS. It will be natural for ESRI people, for example, to want to use words like "theme" when referring to the drawings in their projects, but that will only get confusing because the words you use won't match what is written in the documentation.
Drawings, Images and Maps

Most files imported from other GIS systems into Manifold will appear as drawings. Most raster data and other images imported into Manifold will appear as images. The drawings imported into Manifold were likely called "maps" in their originating GIS programs.

Within Manifold, drawings, images and other components (such as labels or tables) can be clicked open into windows. Within those windows virtually all Manifold commands, including editing, transforms and more can be applied.

Drawings can be opened within a drawing window. Drawing windows always show the drawing as represented by the native projection used by that drawing. The drawing window above shows a drawing called bay_hydro that shows hydrography lines for the San Francisco Bay region.

Images can be opened within an image window. Image windows always show the image as represented by the native projection used by that image. The image window seen above shows the SanFran image that is a Landsat photograph of the Bay area. Images may be grayscale, palette, RGB, RGBA, or compressed images. Many image editing commands work only with RGB or RGBA images. See the Image Types topic.
Maps are made up of layers where each layer is another component in the project. Opening a map in a map window will show the components within the map in layers with a layer tab for each layer. Maps can use any projection, even projections that are different from the components they contain. The map illustrated above uses the bay_hydro drawing and the SanFran image as layers.

Important: For different layers to be seen together in a map window in correct alignment they all must be correctly georegistered. If the drawings or images used in a map have been imported from a geographically-aware format that saves projection information within the format, then there's nothing to think about and we can use such drawings and images together right away. On the other hand, if the drawings and images are stored in geographically-unaware formats, such as, say, DXF for drawings and JPG for images, well, then we may have to use the Edit - Assign Projection dialog to tell them what projection they are to use. We may even have to georegister them correctly within Manifold before we can use them together in a map.

Advantages of Maps

Maps have several characteristics that set map windows apart from drawing or image windows:

- Maps can consist of an unlimited number of layers where each layer is a drawing, labels, theme, image or surface component. Drawing and image windows consist of only one component.
- Context menu commands for layer tabs in maps allow us to create, duplicate and otherwise manage components in the project for use as layers in the map.
- Maps can be re-projected on the fly to present a different appearance of their contents without actually changing the data of the components within them. In contrast, drawing and image windows always show their contents using the native projection of the data set they contain. Changing the projection seen in a drawing or image window changes the coordinate data within the data set.
- Maps can alter the Layer Opacity of the layers in them to achieve different visual effects.
- Maps can use Zoom Ranges to automatically turn different layers off and on for display as the zoom level changes.
- Maps provide a user interface for commands that involve more than one layer at a time, such as the Match function, the Transfer Selection command and Tracing between layers.

Using many layers allows us to compose maps that are visually very sophisticated while retaining simplicity in the building blocks used. Each layer may be independently turned on, off, moved or otherwise controlled. For example, we can change formatting in a drawing layer without affecting other drawing layers. Using labels components we can build up many layers of text and annotations where each different text layer uses a different font, size or color for the text.

For certain casual uses, it may be easier to simply click open a drawing or image in its own window without opening it within a map. At times we will work with a drawing or image in a map and also have it clicked open within its own window. Within the map window we can see the drawing or image in a combined layer stack together with other components. Within its own window we can see and work with the image on its own.
This is also a great way of resolving visual ambiguity or making selections when working with many layers of complex drawings in a map. We can always click open a particular drawing in its own window and make a selection there in addition to seeing the drawing in a map and working with it in the map as well.

Maps and the components they contain as layers are dynamically linked. If we change something in a drawing's layer in a map, that change will instantly apply to the drawing open in its own window as well and vice versa.

Advantages of Component Windows

Opening a component in its own window does have some advantages.

- Component windows provide the very fastest interface in Manifold, which can be handy when working with very large images or surfaces that use a projection different from a map that might include them as a layer. However, in most cases when working with large images or drawings users will take care to assure that any map that includes them as a layer uses the same projection as well. Doing so will provide performance in the map window that is effectively the same as in the component window.
- Because the component is always shown in its native projection the exact relationship of objects within their native coordinate system is always evident. If a map uses a projection that is very different from that used by a component the change in visual appearance when seen through the map's projection might be disorienting.
- Certain commands must operate within the component window. For example, zoom ranges must be specified within the component window because otherwise the component would be invisible at certain ranges when seen within the map window.

Surfaces and Labels

The comments in this topic apply to themes, surfaces and labels components as well as to drawings and images. Themes, surfaces and labels components can be opened in their own windows as well as being used in maps.

Labels components, in particular, tend to be used mostly as layers in maps and are rarely opened in their own windows. Drawings by themselves can only show points, lines or areas. To show other visual elements, such as labels, we use a labels layer together with a drawing layer in a map.
Working with Maps

Maps show drawings, images, surfaces or labels as layers in a map window. Maps are the primary user consoles in Manifold because they allow us to work with many layers, and thus many components, at once. For example, we can edit drawings by working in a drawing layer to trace over items seen in an image layer below.

Maps can show their contents using any desired projection no matter what the native projections might be of their constituent layers. Maps work faster if they use the same projection as their constituent components, but they are not required to do so.

Creating a New Map

We can create a new map using whatever source materials we like:

1. Create a new project using File - New. This opens a project pane.
2. Using File - Import, import any drawings and images into the project the new map will use.
3. Create a new map in the project using File - Create - Map. We must specify the name of at least one drawing or image this map will use. In the Create Map dialog, check all the drawings or images from the project you would like to appear in this map.
4. Double click on the new map in the project pane to open it in a map window.
5. Drag and drop any additional drawings and images from the project pane into the map window. This adds them to the map. We can also add a component from the project by right clicking on any existing layer tab and choosing Add - Layers from the context menu.
6. Add a new blank drawing to a map by choosing Add - New Drawing. This will also create the drawing in the project.
7. If images or drawings are not already georegistered, they must first be georegistered so that they appear in the proper geographic location at the correct size.
8. If desired, change the projection used by the map.

Note: As a convenience, if a component is highlighted in the project pane when the Create Map command is launched it will automatically have that component checked for use in the new map.

Projections

When a map is first created in a project it uses the projection of whatever component(s) it uses. If it was created using only one component the projection of that component will be used for the map.

If a map is created using multiple components at the same time, then Manifold will use the projection of the largest image or surface as the default projection. This is because images and surfaces are slower to re-project than drawings so the map will run faster if it takes the native projection of an image or surface and then re-projects any drawings to match.
If only one image or surface is used to create the map then the projection of that image or surface will be used. If several images or surfaces are used to create the map then the projection of the largest of those images or surfaces will be chosen as the initial projection of the map.

We can change the projection used by a map by using the Edit - Assign Projection dialog. This dialog changes the projection used to display the map without changing the projections used by the components that are layers in the map.

If we would like the map to use a specific projection that's used by one of the components it shows we can easily specify the projection to be used by right-clicking on any layer and choosing the Use Projection choice from the context menu. The map will automatically switch to using the native projection used by that layer.

On the other hand, if we would like to re-project a component to use the same projection that is used by the map, we can right-click on the layer tab for that component and then choose Project to Map.

**Layers**

Maps show components stacked in layers.

The "white space" in drawings, themes and labels components is transparent so objects in lower layers can be seen. Images and surfaces can also contain regions of invisible pixels that allow items on layers below them to be seen. Images can also use RGBA pixel transparency to provide continuous gradients in transparency on a per-pixel basis.

Any layer may also have transparency specified for the entire layer via Layer Opacity settings. For example, we might show a surface layer in partial transparency over an image layer that provides a background image and so use the surface layer to provide a sense of "hill shading" and thus a sense of depth to the image to give it greater visual impact.

Each component dropped into a map appears as a layer identified by a layer tab.

Maps show layer tabs at the bottom of each map window. The layer tabs are automatically named the same as the drawing or image they contain. The leftmost tab is the top layer. Layer tabs may be used to manipulate the layers in a map:

- **Click** on a layer tab to make that layer the active layer.
- **Double click** on a layer tab to turn that layer on / off for display.
- **Click and drag** a layer tab to move that layer to a different position in the layer stack. Alternately, right click on the layer tab and use one of the Order commands.
- **Right click** on a layer tab to pop open a context menu of useful commands.
The **active** layer tab is white. Other layer tabs that are being displayed will have default contrasting color. Layers that are turned off will appear in a "disabled" display style. Only one layer is active at a time. New objects are always drawn in the active layer.

Manifold will automatically reconfigure all system controls for correct operation with the active layer. If the active layer is an image, for example, the Transform toolbar will be loaded with image operators. Click on a drawing layer tab to make it active and the Transform toolbar will switch to drawing operators.

When working with many layers, many users will also use the **Layers Pane** to control layers in addition to using layer tabs. See the Layers topic and the View - Panes - Layers topic for more information.

The layers pane also includes checkboxes for two system "layers" - a background color layer and a border layer that shows an enclosing box for all map layers. If any print layouts have been created using the map the layers pane will also include additional "layers" with checkboxes that may be used to show which regions of the map are included in the layout.

**Commands and Layers**

Selection commands and Windows Clipboard commands (**Delete**, **Cut**, **Copy**, **Paste**) work on the active layer only by default. To extend the action of a selection command or a keyboard shortcut Clipboard command to all visible layers hold down the **ALT** key while executing the command. Think of the **ALT** key as a mnemonic for "All" layers.

For example, using **Select Box** in a map window selects objects in the active layer only. Pressing the **ALT** key while using **Select Box** will select objects from all visible layers. The **ALT** key does not affect menu commands such as the **Modify Selection** commands in the **Edit** menu or the Clipboard commands in the **Edit** menu.

**Layer Restrictions**

The **Restrictions** button in the layers pane displays and sets **layer restrictions**. Layer restrictions may also be set by right clicking on a layer tab in a map window and choosing **Restrictions**. Each layer within a map may be specified to be clickable, editable or selectable (in any combination) with mouse commands. The restriction specified operates only on mouse commands. Other commands such as keyboard shortcuts, menu commands or query toolbar commands are not affected.

Layer restrictions are used to show layers in maps while preventing some layers from participating in mouse commands. This makes it easier to edit desired layers in complex maps. Another use for layer restrictions is in Manifold IMS to restrict the action of hyperlinks or the **Info** tool to a limited set of layers.

**Adding a New Layer**

Manifold provides several ways to add layers so that the most convenient method is always at hand:

- Drag and drop a component from the project pane into the map. This is a fast way to add just one component. The layer will be dropped into place just above whatever was the active layer.
- Right click onto any existing layer tab and choose **Add - Layers** from the context menu. A dialog with checkboxes for all of the currently unused components in the project will pop open. Check one or more boxes to add one or more components as layers just above the active layer. This is a fast way to add more than one component. It is also a fast way to add just one component if the project pane has been closed (say, to maximize screen real estate available for the map window).
- Right click onto a layer tab and choose **Add - New…** to add a new drawing, image or labels component. This creates a new, blank component of the desired type in the project and inserts it as a layer above the active layer. This is a fast way of creating a blank labels component to insert labels manually, or to insert a blank drawing to which items from other drawings may be pasted, etc.
- Right click onto a layer tab and choose **Duplicate**. This makes a copy of the active component (that is, the active layer) in the project and inserts it as a layer above the active layer.

**Zoom Ranges**
We can tell Manifold at which zoom level a layer becomes visible or invisible by using zoom ranges. Clever use of zoom ranges can set up a map so that as users zoom into the map different information comes into view. See the Zoom Ranges topic for details and examples.

**Transparency / Opacity in Layers**

Any part of a drawing or theme that is not occupied by an area, a line or a point and any part of a labels component not occupied by a label is transparent. This normally appears as "white space" because the default background color of Manifold windows is white. It is possible to use white as a background (fill) color for areas, in which case it is a good idea to change the background color to something other than white. If this is not done, it will be too difficult to tell what regions are areas and which regions are simply empty regions bounded by lines.

Images and surfaces have no empty space. Every part of an image or surface is filled with pixels. To allow "see through" regions in images and surfaces, Manifold images may have invisible pixels (which are used to show regions of missing data in surfaces) through which any items in lower layers may be seen. Invisible pixels are simply placeholders that do not appear. It is as if there are no pixels in that part of the image. To make part of an image transparent, we select the desired region and then Delete those pixels. Invisible pixels may be used with any of the four types of images.

**RGBa** images can have each individual pixel assigned a percent pixel opacity value using **RGBa Pixel Transparency**. This is normally accomplished through masks and other tools. When combined with Manifold image editing tools this effect can be used to compose amazing images by combining many layers of other images. See the Layers topic for an example.

Any layer can be made partially transparent by changing its opacity from 100% to 0% in steps of 1 percent by right clicking on the layer and choosing **Opacity**, or by using the controls in the Layers pane. This is a great way to create spectacular effects. Transparent layers work with all types of components that may appear in maps. See the Layer Opacity topic.

To summarize, there are several types of transparency that may be found in a map:

- Drawings and themes are transparent in **empty space** not occupied by an area, line or point.
- Labels components are transparent in **empty space** not occupied by a label.
- Any image or surface may be made fully transparent in regions of **invisible pixels**.
- **RGBa** images may be made partially transparent using **RGBa pixel transparency**.
- Any layer may be made partially transparent using **layer opacity**.

**Selection in Maps**

Selection in maps works the same as it does in all other windows, except that by default only items in the active layer are selected. Anything that is visible may be selected with mouse selection methods such as **select box**, so long as enabled by the selection modes in use. To make selections in more than one layer at the same time, hold down the **ALT** key.

For example, we can draw a box that encloses both lines in one drawing layer and areas in a different drawing layer. If the active layer is the lines layer, only the lines will be selected. If we hold down the **ALT** key while drawing the selection box with the mouse, both the lines and the areas will be selected.

**Multiple Map Windows**

We can open more than one window for the same map. Except for independent pan and zoom, each window into a map is perfectly synchronized with all other windows into that same map.

- Each map window may be resized, panned and zoomed independently of the other windows into that map.
- Changing layer tab order or visibility in one window changes it in all windows into that map.
- Changing the projection used in one map window changes it for all windows into that map.
- Any changes in drawings and images in one map window will appear in all windows into that map.
- Any selection made in one map window will appear in all windows into that map.
To see the map in a different window or with different layer ordering, make a copy of the map in the project pane and then change the projection or layer ordering in the copy.

**Map Windows are Synchronized with Components**

Maps are simply views into more than one component at once. Any changes to a component within a map window will appear throughout all other windows in which that component is seen. Components may be opened in one or more native windows at the same time they are seen in one or more map windows. For example, a drawing might appear in two drawing windows (each of which could be zoomed into a different part of the drawing) at the same time it appears in several different map windows.

- Any changes made to the component in any window will simultaneously appear in all other windows showing that component.
- Any change to the component in its layer in the map window will also instantly appear in any of its component windows as well.
- Any selection made in the component will be shown in all other windows where the component appears.

Note that re-projecting a component within its native coordinate system will change its appearance in any native component windows in which it appears but will not change its appearance in any map in which it appears. There is no apparent change because maps always re-project on the fly all components into the projection specified for the map regardless of what native projection is used by the component.

**Maps can use Projections**

Projections are a way to provide realistic views of the Earth, a 3D sphere, on 2D surfaces like paper maps and monitors. Projections are also used in paper maps to make measurements. It comes as a surprise to many people that there is no way to draw large regions of the Earth on a 2D surface without distorting what is shown. Geographic projections provide a standard way of presenting the features on the Earth's surface in 2D form that minimizes distortion in desired ways.

An unprojected map of the Earth may be shown using the Latitude / Longitude projection, which plots the features of the Earth as if latitude degrees were Y coordinates and longitude degrees were X coordinates. This presents a highly distorted representation of the Earth in higher latitudes, since obviously Antarctica does not have as much land area as Eurasia and Africa together.
Maps in Manifold can automatically show their contents using whatever projection is desired. All layers used in the map, including any image layers or drawing layers, will be displayed in the desired projection. For example, the illustration above shows the Earth as projected using the Hammer projection.

When a projection is changed in a map there is no change in the actual data of the components involved. The new projection is simply a change in the way the map window displays the components. Components such as drawings or images might be in their own projections that are different from the projection used by the map. The map window will dynamically (“on the fly”) recalculate projections used by components in the map so the map can display all layers together in the desired projection.

Users experienced in GIS know that re-projecting a drawing or an image changes the data. Many re-projections will slightly reduce accuracy. Changing a projection in a map in Manifold does not have this effect because the original data in the drawings or images used in the map is not changed: only the view of the data as seen through the map window changes. This allows us to change projections in a map as often as we like.

Different projections have different characteristics and are usually selected for a particular purpose because they minimize a particular type of distortion at issue in the subject matter of the map. The Hammer projection seen above, for example, distorts the shapes of items on the periphery of the projection but it maintains the relative area of each feature. It is a good choice to provide a better impression of the relative size of Antarctica compared to other continents in the context of an entire world view. However, it is a poor choice to portray the shapes of North American features as they would be seen on a globe.

No one projection can serve all purposes equally well. The topics in the Projections chapter discuss projections and provide a guide to choosing a good projection for one's intended use of the map.

Projections are a more sophisticated topic than can be covered in this introduction. Please see the Projections topics for information about geographic projections and their use in Manifold.

**Matching Map and Component Projections**

Maps can work faster if the drawings and images they contain as layers use the same projection that is requested of the map. If we always use a particular projection within a map it will be worth re-projecting larger drawings or images into that particular projection as well. To permanently re-project a layer in a map to match the projection used by a map, right click on that layer tab and choose *Project to Map*. This will automatically re-project that layer into exactly the same projection used by the map.

In order for maps to run faster when the constituent drawings or images are in the same projection, they must use exactly that same projection in all parameters. It does not help if different components use Lambert Conformal Conic projection but with different datums, for example. Most users will choose a particular projection and parameters they like and will use that as their standard for a given subject area.

The *File - Create* dialog takes the projection parameters for the new component from whatever window is active at the time the new component is created. If no existing component is active, the new component will be created using the system default projection of Orthographic centered at the 0,0 world latitude/longitude origin. If a map window or other component window is active when the new component is created, the new component will be created using whatever projection is used by that active window. This context-sensitive setting of default projection parameters makes it much easier to create new components using projection parameters that are hassle-free by default.

In particular, if the map we are working with has the focus when we use *File - Create* to create a new, blank component we will assure that the new component matches the map's projection for good performance. If we create a new component using the layer tab context menu's *Add - New...* commands we will automatically create a new, blank component that matches the maps' projection.

Another way of making sure a map matches a particular component's projection is to create that map based on the component. Right click on the component in the project pane and then choose *Create - Map* and check the component's box in the map creation dialog. In fact, when creating a map that will incorporate some large images or drawings, it is a wise idea to first create the map based on the biggest image that will be used.

**Maps are Free**

A "map" in the project contains very little data, since the contents of the map are taken from the components it contains. The actual map component consists of pointers to the relevant components as well as a very small amount of housekeeping information (layer ordering, projection parameters, etc). Therefore, making copies of
maps or making lots of maps in a project is an efficient way of creating and saving different views into the same data. There is virtually zero storage cost to adding another map to a project.

**Maps in CAD and Image Editing**

Manifold uses the word map to name the component that shows other components in layers and that can display its contents in whatever geographic projection is desired. We use the word map because displaying drawings and images in a geographic projection is quintessentially the function of a "map."

However, if we set aside the idea of projections what we have left is a viewing window that can show and manipulate components in layers. That's exactly what is done in the main viewing windows of CAD editors like AutoCAD or graphics editors like Adobe PhotoShop or Adobe Illustrator. It's just that ordinary CAD editors like AutoCAD or image editors like PhotoShop lack the projection and other geographic capabilities of Manifold maps.

Layers are a very important part of maps (and of the analogous viewing windows in AutoCAD or PhotoShop) because they allow us to compose very complex visual effects by combining images and drawings in layers. They also allow us to keep different types of vector objects within different drawings (for a more orderly workspace) and to re-use images or drawings in different combinations.

Layers are a great way of creating spectacular visual effects by combining multiple images. Through the use of pixel transparency, layer opacity and invisible pixels we can manipulate and position pixels within stacks of image layers in an almost magical way. We can also use Manifold as a superb CAD editor by working with drawings in the same way. Finally, we can combine drawings and images to create complex and visually elegant documents for presentation purposes.

The layer structure of Manifold maps was originally intended to make possible the creation of any sort of geographic map. As the image and drawing editing capability of Manifold grew it has now become possible to use Manifold "maps" as a framework for fairly sophisticated image and CAD drawing editing. In this sense, Manifold "maps" are much more than just geographic maps: they are general purpose, multi-layered documents that may be used in image editing and CAD editing as well.

**Tech Tip**

Although maps are free in that copying and pasting a map won't make duplicate copies of the components within that map, when a map is copied Manifold will remember the layers that are supposed to be in that map until it is pasted. So, for example, if we Copy a map and then before it is pasted we delete one of the components that is a layer in the map from the project pane, if we then Paste the map the deleted component will be restored.
Working with Large Images or Surfaces

Manifold can work with immensely large images and surfaces. Because the maximum size of images and surfaces is so large (16 exabytes, that is, 16 trillion gigabytes), as a practical matter the size limitation imposed on images or surfaces will be the amount of free disk space available or the speed with which a given machine can process the tasks desired. As images or surfaces become very large operations involving them will slow down because more pixels must be processed.

See also the Performance Tips topic to maximize performance in this or any other Manifold operation.

Manifold has several strategies to increase the speed with which large images or surfaces may be manipulated:

- Use of compressed images - Compressed images will load and display much faster than other types of images. The cost of using compressed images is that they are "display only" and cannot be modified.
- Use of intermediate views - When working with all non-compressed images and all surfaces Manifold can use pre-built intermediate displays to increase the speed of redisplay, panning and zooming. The cost of using intermediate views is that the first time an image or surface is displayed the system will pause while intermediate views are constructed.
- Use of linked images from image servers - Linked images usually store their data outside of Manifold, which allows using highly scalable external sources for both the actual image data and also intermediate views. Linked images are frequently "display only" and cannot be modified. However, when using an image server accessed through the Internet or other slow network keep in mind that the slow speed of the network may prevent fast operation of linked images from the image server.
- Use of linked images from Oracle GeoRaster capable servers - Oracle's GeoRaster technology, available within some Oracle products such as Oracle Spatial and Oracle Enterprise Edition, is extraordinarily fast. Linking a drawing from such an Oracle server allows working with images that up to terabytes in size.

Compressed Images

Compressed images use sophisticated wavelet compression technology to not only compress the amount of data an image requires but also to reconstitute the image on the fly on demand. At any given zoom level the desired view of the image is reconstituted from the compressed data store. At the present time, Manifold supports use of compressed images using ECW (the same as used with ERMapper) and JPEG 2000 formats, the emerging open source standard.

Compressed images can be either stored within a Manifold project or linked into the project as a linked image from a compressed format file stored outside of the project.

This technology allows astonishingly fast loading and display of such images but this speed is achieved at the cost of significant limitations. The main limitations of compressed images are:

- Display only - Compressed images may be viewed but not manipulated.
- No selection - Compressed images cannot be altered and their pixels may not be selected.
- Limited projection systems - Compressed images may be viewed only within a map that has a non-curvilinear coordinate system. As a practical matter, maps that will display compressed images should be created from those images.
- Reduced information content - The compression process eliminates the original pixel information in the image. Although compressed images when reconstituted for display may appear visually identical to an original uncompressed image, the reconstituted pixels are a synthetic approximation of the originals.

Compressed images also provide significant benefits:

- Fast display - The reconstitution process provides for very rapid recomputation of an image view at any desired zoom or pan.
- Fast loads and saves - Projects containing compressed images load and save much faster than projects using uncompressed images.
- Linked images - Compressed images may be left in their original .ecw files and, instead of being imported into the Manifold .map project file.
Introduction

- **Reduced disk space** - Compressed images require much less room than uncompressed images. Using linked images to utilize a single image file in many different projects will also save much disk space as compared to duplicating that image within many projects as an imported image.

- **Streaming links** - Compressed images may be linked into a Manifold project from an ECWP server URL, loading from a data stream served by an ERMapper Image Web Server.

- **Multi-channel images** - Some types of compressed images can store many more channels than just three channels that are always used for R, G and B channels. The *Image - Display Options* command in Manifold allows selection of which channels from a multi-channel image will be used for R, G, B and alpha channels.

Compressed images are a good choice for images displayed for their visual merit only, such as when images are used to form a background layer for a map, or when a background image layer will be traced to create a new vector map. Compressed images, especially those that are linked into the project, provide rapid response even when the image data is tens of gigabytes in size.

**Intermediate Views**

For simplicity of language this topic discusses intermediate views in the context of images; however the following applies to surfaces as well.

Manifold has several strategies for working with very large images to improve interactive performance and to reduce the patience required of a user. The most important strategy is building intermediate views of an image in advance and storing those intermediate views invisibly along with the image for future use. The intermediate views are then used to display the image for faster panning and zooming.

To understand intermediate views, consider the case of an image that is 21600 x 43200 pixels in size. Such an image is about 3.7 gigabytes in size if it is an RGB image.

A 21600 x 43200 image cannot be seen in full resolution on a computer monitor because there are not enough pixels on even a 1280 x 1024 monitor to show the entire image in full resolution. When a very large image is zoomed out so the entire image can be seen what is displayed is an interpolation, like a large thumbnail, that summarizes the vast number of pixels in the real image within the relatively few 1280 x 1024 pixels available on the monitor.

As we zoom further into the image we see it at resolutions that are closer and closer to the real resolution. When we zoom far enough into the image so that one image pixel uses one pixel on the monitor we see it in "native" resolution, but we only see a small part of the image (about a thousandth of it) at a time. Every time we show a very large image at less than full resolution the system must compute an approximation that fits into the view window size being used. The process of computing a correct interpolation at various zoom levels to smoothly zoom and pan the image can require a lot of time.

Manifold shortens this time by computing in advance a series of intermediate views of the large image for use at intermediate zoom levels and saving those intermediate views as a built-in part of the image. The intermediate views allow much faster action than if the entire large image must be manipulated for every change of view or other action.

When the image is saved as part of a *.map* file, the intermediate views will also be saved with the image. The next time the *.map* file is loaded and the image is opened the image will display very rapidly because the intermediate views are already available.

The use of intermediate views allows much faster action when working with large images but at the cost of a slower "startup" when the image is first created and opened. After importing a large image (which can take a considerable period of time for very large images) the very first time the image is opened a lengthy pause will occur as Manifold computes and saves intermediate views. After that, subsequent actions with the image will be very fast. When an image is saved as part of a *.map* file the intermediate views will be saved within the *.map* file as a built-in part of the image.

Whenever a map file is loaded and an image in the *.map* file is opened it will open instantly, because the intermediate views are already ready for use and are cached within the *.map* file. The use of intermediate views therefore makes it profitable to keep very large images saved in *.map* files.

Except for the delay in opening an image when it is first created or imported, there is no impact on the user of intermediate views except for the much faster performance they make possible. They are a completely "built in" part of Manifold that is handled automatically within the program with no user intervention required.
Many other technologies also incorporate the idea of intermediate views. See the Intermediate Levels and Pyramids topic for additional discussion and examples.

**Linked Images**

**Linked images** are images that are created using information from database tables or which are created using data streamed into the project from an image server. Linked images can also be used to reap the benefits of compressed image technology as discussed above.

The main advantage of linked images is the potential for great display speed of images that can, in some cases, be virtually unlimited in size. In addition to the potential for great speed, a secondary advantage of linked images is that multiple users can use the same image at the same time in different projects.

As far as disadvantages go, the main disadvantage is that in almost all cases linked images will be display only and **are subject to the same limitations as compressed images**.

Linked images from image servers are a way of bringing images into Manifold from external servers. The advantage of such linked images is that we can use data from immensely large image servers that save terabytes of information and which might be using hundreds or even thousands of machines in parallel to serve the views we want. Images linked from image servers will be displayed using a mosaic of image tiles that are automatically downloaded from the image server and seamlessly put together for us by Manifold to show the view we want.

The process can be remarkably fast because for any given level of zoom or panning into an image the tiles that comprise that view will be composed for us by the image server and then tossed down the network connection into Manifold. All Manifold need do is compute what tiles at what zoom levels are needed and then tell the image server what to send. The extraction of data for the tiles and the composition of tiles will be done by what is likely to be a very fast group of machines.

The main disadvantage of using image servers is that whatever views we want to see must be sent to us via a network connection to that image server. If we are using some public image server like TerraServer, each tile must reach us via what is often a rather slow Internet.

A further disadvantage of using image servers is that although public image servers like TerraServer or Google Earth will often use extraordinarily fast arrays of machines, all that fast hardware is very often overburdened by an extraordinarily large number of users. Response, therefore, can be slow.

Finally, when we link in what might be a very large image from an image server we will not likely ever get all of the data for that image from the image server unless we do a deliberate download at full resolution of the linked image into an ordinary local image.

Linked images created from database tables can be created using any database technology, but usually in the case of very large images they are created using Oracle database products that incorporate Oracle’s **GeoRaster** technology, such as Oracle Spatial.

**Linked images from Oracle servers** can be breathtakingly fast even for immensely large images. Even when an ordinary desktop machine, such as that using an entry-level 64 bit AMD processor, is used to host both the Oracle server as well as running the Manifold session, even gigabyte sized images linked from that server will zoom and pan virtually instantly.

The main advantage of storing images as linked images within Oracle databases using **GeoRaster** technology is that Oracle’s technology scales very well. Oracle databases are very well suited for clustering architectures that enable creation of very large, very fast Oracle servers. In such cases, images of virtually limitless size, even terabytes, can be stored and displayed with great speed. Even in relatively small and inexpensive machines the size of images that can be stored within Oracle is truly impressive.

The disadvantages of storage within Oracle **GeoRaster** capable databases are cost and some additional administrative overhead. As of this writing, freely redistributable versions of Oracle Express Edition do not include **GeoRaster** capability, so one must acquire a suitable Oracle product. That is not normally an obstacle as usually those applications that involve many gigabytes or terabytes of image data have sufficiently high value that the cost of requisite Oracle licenses is a small part of the overall budget.
**Note:** Manifold also allows storing images within database tables in databases other than those using Oracle GeoRaster technology and then reconstituting those images as **linked images** created from those database tables. However, image storage within databases other than Oracle GeoRaster storage has relatively poor performance and is not usually done except for specialized applications where fast performance or very large image size are not required.

### Image Size and Compressed Formats

Some formats, such as **TIFF** or ordinary **JPEG** will use compression to reduce the amount of disk space required for the image even if the format does not otherwise use compression or intermediate level strategies to increase display speed as do more sophisticated formats like **ECW** and **JPEG 2000**. Whether or not compression is used for speed, just as a matter of saving disk space very large images are almost always saved in some compressed format.

Because of the limits of what a particular format can do, Manifold must use different strategies when importing images from different formats. Most compressed formats, such as **TIFF**, will be automatically decompressed upon import into a Manifold project. **ECW** and **JPEG 2000** format compressed images are not decompressed but are imported into Manifold as a special compressed image type. The difference between decompressing an image upon import and not decompressing an image can be substantial.

For example, a 21600 x 43200 pixel image requires about 3.7 gigabytes as an RGB image but it might be saved as only a 400 megabyte file in .tif format. When such images are imported into Manifold they must be decompressed into a real image to be ready for display and editing. That can come as a surprise if we only look at the size of the compressed .tif file on disk and don't consider how big the image will be when decompressed on import into the Manifold project.

Because a 400 megabyte .tif file can easily decompress into a 3.5 gigabyte RGB image it is essential to have plenty of free disk space when working with large images. If we work with images that are hundreds of megabytes in size when saved in compressed formats like .tif, we should have gigabytes of free space on hard disk, preferably dozens of gigabytes if we will be working with several such images at once.

In contrast, images stored within .ecw format will be brought into Manifold with their compression retained. They will not expand in size because they will be brought into Manifold (either imported or linked) as **compressed** images.

Likewise, **linked** images of all kinds that take their data from outside of the Manifold project will not have high storage requirements upon import. The price we pay, of course, for such reduced storage requirements is that such images are display only and cannot be edited, selected or otherwise manipulated.

### Disk Utilization

Because large images and surfaces that are imported will not usually fit into system RAM, they will normally be kept on disk, either in the .map file or in temporary disk files created when the .map file is loaded. Whenever a large image or surface is opened Manifold will first conduct a rapid calculation to check the amount of RAM available and to determine the optimum way to balance usage of RAM with disk space for working with a particular image.

Obviously, since RAM is cheap if we are working with large images it makes sense to install the maximum amount of RAM we can into our systems. If our system can host four gigabytes of RAM then we should install four gigabytes.

With larger images unless compressed images or linked images are used just about any operation with such images will inevitably involve using large amounts of disk space, whether it is as a result of using the system pagefile or the use of fast, temporary files set up by Manifold itself on disk. Of course, if the images are to be manipulated in any way (other than just merely displayed) compressed images or linked images cannot be used.

This heavy usage of disk for large images and surfaces makes it imperative that users desiring optimum performance with large images should invest into fast hard disks. A good strategy is to buy large hard disks, since larger hard disks are faster than smaller ones, to choose faster-rotating disks such as 7200 RPM or faster to reduce rotational latency and to invest in SATA or other fast interfaces. Using multiple hard disks in striped RAID arrays can also increase disk performance. Although most users will not invest in elaborate RAID configurations, simply adding a second hard disk to host the TEMP folder separately from the system pagefile will provide some performance gains because the two disks will be able to independently perform overlapping seek operations.
Tech Tips

Expert readers may wonder how much space is used up in the .map file to cache intermediate views. The usual rule of thumb is that the compressed size of the image within the .map file will increase by about 30%. Although that seems like a lot, considering that the overall compression of the image will continue to be somewhere between a factor of three and ten, it is not a bad tradeoff to achieve much faster panning and zooming.

Keep in mind that there are not enough pixels on the screen in computer monitors to display all of a large image at once. When zoomed out so the entire image can be seen the large image will be shown as a summary, intermediate view using fewer pixels. The summarization process necessarily loses detail. This is not a limitation of Manifold but rather the result of a limitation in the resolution of the computer display. If the image is printed at full resolution (inkjet printers can now routinely provide real resolutions of over 1400 or 2800 dots per inch) it will appear much sharper and clearer than it can appear on screen in an intermediate view.

Manifold uses the unlimited size ECW SDK to support use of ECW and JPEG 2000 images of unlimited size for all loading, linking, saving and display operations. Images may be converted to and from compressed mode using the Image - Convert To command. Keep in mind that converting a compressed image into another image type, such as an RGB images, can vastly expand the size of the image.

If images are to be used for display only, it make sense to store them in the form of ECW format files and to link them into a Manifold project as compressed images. That's a highly effective and free strategy for working with images up to several tens of gigabytes in size.

In contrast, if images must be manipulated in some way (for example, for editing), then they should be stored in some other format and imported into Manifold as an RGB or other non-compressed image type.

For extremely large images in the hundreds of gigabytes or terabyte range the best strategy is to use an Oracle server cluster with Oracle Spatial or Oracle Enterprise Edition with GeoRaster capability. Oracle servers are also the preferred strategy if very many users must simultaneously access the linked image. For example, if we have thousands of Manifold users in our organization that need to use parts of a particularly large image as a background it is wise to host that image on an Oracle server.

See Also

Performance Tips
Selection

Many operations in Manifold begin by selecting something and applying an effect to that something.

The "something" might be a region of pixels in an image, a particular group of lines in a drawing or some special selection of records in a database table.

The "effect" could be an operation that transforms what’s in the selection, such as brightening only those pixels selected in an image or it could be a request to analyze what’s in the selection and make a report. For example, we might select a group of records in a database table and ask what the average is of the "Population" field for those records.

Choosing that "something" on which we wish to work is making a selection. There are two key ideas connected to selections in Manifold:

- **Making a Selection** - Use the mouse to point at something or to drag open a rectangular, circular or free-form box to select everything inside the box. Selection Tools let us do this. We can also make selections with the transform toolbar, with scripts, using the Query toolbar or from SQL.
- **Changing the Selection** - After we’ve made a selection we may want to add to it, subtract from it or otherwise change it. This is what selection modes and selection commands help us do.

The examples below are illustrated using images. The concepts and methods they show work with drawings as well. To follow along, use File - Import - Image to import the bronze.jpg sample image into the project. Double-click on the bronze image component to open it in an image window. The bronze.jpg image may be found in the Manifold examples downloads.

**Note:** Selection can only be done in components that are read/write. Selection is not supported in images or drawings that are read-only. For example, Compressed images are a special type of read-only image that may be displayed but not otherwise manipulated. Compressed images do not support selection.

**Making a Selection**

1. Choose the selection tool. We’ve picked the Select Box rectangular box selection tool.

2. Make the selection by clicking and dragging open a box with the mouse.
3. The selected pixels will be shown in red selection color.

**Note:** we can change the display style used to show the selected region so that the red crosshatch pattern does not get in the way of graphics editing. This is discussed further down in this topic.

Other tools allow making the selection in different ways. In the image above we've used **Select Touch** to click on the blue sky and thus selected all blue pixels within the touch selection’s tolerance. Following is the full set of mouse selection tools available in Manifold:

- **Select Touch** - Click to select pixels like the one clicked, give or take the tolerance that may be set in the Tool Properties pane. See the following note on touch selection in surfaces.
- **Select Shape** - Left click to draw a closed shape and right click to end and select all pixels within the shape.
- **Select Freeform** - Select objects within the mouse cursor region defined by clicking and dragging, followed by a right click.
- **Select Box** - Click and drag to select items within a rectangular selection box
- **Select Box on Center** - Click and drag open a selection box that’s always centered on the point of first click.
- **Select Circle** - Click and drag open a selection circle within the rectangle implied by the mouse motion.
- **Select Circle on Center** - Click and drag open a selection circle that’s always centered on the point of...
first click.

- **Select Ellipse** - Click and drag open a selection ellipse within the rectangle implied by the mouse motion.

- **Select Ellipse on Center** - Click and drag open a selection ellipse that’s always centered on the point of first click.

**SHIFT key**

Shifts selection to "open" mode. In drawings, selects all items any part of which is within the shape drawn with the mouse.

In images (using **Select Touch**) the SHIFT key will apply the touch select to all pixels like that one touched even if they are not contiguous.

---

**Touch Selection**

Touch selection in drawings will select the objects touched. In images and surfaces, touch selection will select pixels within the tolerance setting in the Tool Properties pane. We can also select ranges of elevations in surfaces using the transform toolbar. See the Editing a Surface for Visual Effect topic for an example.

**Regular and On-Center Commands**

Note that some commands, like **Select Circle**, also occur in an "on-center" version such as **Select Circle on Center**.

Like selection tools, painting tools occur in both on-center and regular versions. We can see the difference between the regular tool and the on-center tool by painting a circle. Painting tools are used in these illustrations with tool opacity of 50 (half-transparent painting) to better show the difference in operation between regular and on-center commands.

Using **Paint Circle** to click near the center of the monument above and then dragging to the spot shown will open a preview circle as shown. Releasing the mouse button paints within the indicated preview circle. With regular commands, the mouse shows the desired diameter of the circle.
We can use **Paint Circle on Center** to click at the same beginning location and then drag to the same ending location. Note that the preview circle is larger than the previous circle. With on-center versions of commands the initial click is the center of the circle and the mouse motion shows the desired radius of the circle.

**Select Objects Buttons**

Select objects buttons allow us to specify what types of objects are selectable. This is very useful when working with complex maps with lots of objects in sight, because it allows us to enable selections only for certain items, such as pixels. These buttons work like filters: what's pushed in is allowed and what is not pushed in is not.

![Select Objects Buttons](image)

Choose the type of items that are to be selected by pushing the selection filter buttons for the desired types. The buttons above show selection enabled for points and disabled for areas, lines and pixels. If we drew a selection rectangle over a region in the map that contained all four types of items, only the points within the rectangle will be selected. Areas, lines and pixels within the rectangle will be ignored.

- **Select Areas** - Enable selection of areas by mouse selection tools.
- **Select Lines** - Enable selection of lines by mouse selection tools.
- **Select Points** - Enable selection of points by mouse selection tools.
- **Select Pixels** - Enable selection of pixels by mouse selection tools.
- **Select Text** - Enable selection of text labels by mouse selection tools.

All five filter buttons will be displayed and may be used in maps. When working in drawing windows, the **Select Pixels** and **Select Text** buttons will not be visible since there are no pixels or text labels in drawing windows. When working within image windows, there are no select objects buttons displayed since images consist only of pixels. In images, selection of pixels is always enabled.

**Selection Modes**

![Selection Modes](image)

The selection mode buttons control how a new selection made with a selection tool is combined with any existing selection. One selection mode button is always pushed in.
Select Replace - Replace the existing selection with whatever is now selected.

Select Add - Add whatever is now selected to the existing selection.

Select Subtract - Subtract whatever is now selected from the existing selection.

Select Invert - Deselect whatever is now selected that was already in the existing selection and otherwise add what has just been selected.

Select Intersect - Select only the region of overlap between what is specified by the tool and what is in the existing selection. If there is no overlap, nothing will be selected.

The default is Select Replace, where any new selection replaces the previous one.

With Select Replace on, if we click several times using Select Touch we would get a new, different selection each time depending on where we clicked.

If we use Select Add we increase the selection every time we touch a different region of like-colored pixels.

Let's look at a series of examples to illustrate the other modes. In each case we will start with a selected region (made with Select Touch between the horse's legs). A dark circle has been added to show clearly the selection circle drawn with the mouse.
We will then choose a selection mode and use **Select Circle** in a new selection to see what happens:

With **Select Replace**, the circle replaces the previous selection.

With **Select Add** the circle is added to the previous selection.

**Select Subtract** removes the part of the previous selection that was overlapped by the circle.
Invert Selection takes any part of the circle that overlaps the previous selection and removes it from the selection and otherwise adds the circle. If all the pixels in the bronze statue were selected and we wanted to deselect that and instead select all pixels not in the statue, we could use invert selection and draw a rectangular selection box that covered the entire image.

Select Intersect takes the intersection of the circle and the previous selection. Select Intersect is very handy when using different combinations of selection tools and selection modes.

The Selections Pane

Once we pick out a selection, we can save it in the Selections pane. We can then re-use it or recombine it with an existing selection using the Selection Commands. The Selections pane may be popped open from the View - Panes menu. It is often left open and docked into the main Manifold window together with other control and information panes.

Saving a Selection

1. Choose View - Panes - Selections to open the Selections pane.
2. Make a selection.
3. In the Selections pane, click New to save the selection into the list in the dialog.
4. If you don’t like names like "Selection1" and "Selection2" double click onto the selection’s name and change the name.

Shortcuts: Open the Selections pane with ALT-SHIFT-S. Close it with another ALT- SHIFT-S or click onto the x Close button in the upper right hand corner of the pane.
The selection pane is very important when working with the transform toolbar. Saved selections can appear in either the target box or the parameter box.

**Selections Pane Examples**

In the examples that follow, we have used the above steps to save three selections that we have renamed to be called “Sky”, “Clouds” and “Box.”

We can build up the selections quickly by using **Select Touch in Select Add mode**. We could click on different parts of the blue sky, for example, to add all the different blue regions quickly to one selection that we saved as Sky.
Once the above selections have been saved into the Selections pane we can retrieve them at any time.

To see what a saved selection looks like without actually using it, press in the Preview button in the Selections pane. The saved selection will be shown in the drawing window using a blue hatched pattern in addition to any selection that may already be there:

The image above is what the Clouds saved selection looks like as a preview.

This image shows the Box saved selection as a preview.

Selection Commands
Five selection command buttons are arrayed at the top of the Selections pane. These commands apply to whatever saved selection has been highlighted in the dialog. If we click on a saved selection to highlight it and then press one of these buttons, it will be combined with whatever is the current selection in the image. The selection commands allow us to use the saved selection to replace the selection, to add to the selection, to subtract from the selection, to invert with the selection and to intersect the selection.

- **Select Replace** - Replace the existing selection with the highlighted saved selection.
- **Select Add** - Add the highlighted saved selection to the existing selection.
- **Select Subtract** - Subtract the highlighted saved selection from the existing selection.
- **Select Invert** - Deselect what is in the highlighted saved selection that was already in the existing selection and otherwise add the highlighted saved selection.
- **Select Intersect** - Select only the region of overlap between the highlighted saved selection and what is in the existing selection. If there is no overlap, nothing will be selected.

For example, if we click on the Clouds saved selection to highlight it and then press **Select Replace**, the Clouds selection will replace whatever was the previous selection:

This is a common method of retrieving a saved selection: simply click on it in the Selections pane and then press **Select Replace**.

We can also use the selection commands to combine the saved selection with whatever is the current selection in the image. In the examples that follow, we have first made the Clouds selection and then we have highlighted the Box selection which is shown in the preview blue color. After that, we click on one of the selection mode commands to see what happens:
**Select Replace** - The pixels in the saved Box selection are selected and all others are deselected.

**Select Add** - The pixels in the saved Box selection are added to those already selected. Note how the legs of the horse were not selected before and are not selected after the Add operation. Pretty neat, isn’t it?

**Select Subtract** - The pixels in the saved Box selection are subtracted from the existing selection. Note how the new selection has had the region of overlap removed.
Select Invert - Any pixels in the region of overlap are deselected while all those that are in the box or in the previous selection are selected.

Select Intersect - Select all pixels in the region of overlap between the Box and the previous selection. If there is no overlap, no pixels will be selected.

Example: Selection Tools, Selection Modes and Saved Selections

This example shows various selection capabilities used in combination.

Using Select Touch and Select Add we click on different parts of the image to add pixels to the selection as desired. If we "overshoot" and add some pixels we don’t want, we use Select Subtract with a convenient tool such as Select Rectangle to remove the undesired pixels from the selection.
Next, using **Select Intersect** we use **Select Circle** to grab only those previously selected pixels that fall within the circle. We can save this selection into the Selections pane.

When a selection is made in an image, editing effects apply only to those pixels selected. Here, we've used **Hue / Saturation** to change the hue of the selected pixels.

We then clicked **Deselect** to deselect all pixels. Nice, but not bright enough.....

Using the Selections pane, we can recall the selection we used previously. When shown as a preview the saved selection is drawn in bright blue color. We are happy we saved it, since we want to change the saturation and
lightness of exactly those pixels that were in this selection earlier. We press **Select Replace** to use this saved selection in the image.

We've used **Hue / Saturation** to increase the saturation and lightness of the selected pixels and then clicked **Deselect** once more to get rid of the selection hatch pattern. That's better!

**If the Selection Pattern Interferes**

A dense red grid shows the selection well without any ambiguity; however, it can obscure the effect of changes that are applied to the selected pixels. There are two ways to deal with this:

- **Change the selection style to a less intrusive pattern or color or other style.** The pull-down arrow next to the **Use Selection Color** button opens a menu of possible selection styles, including simple border and various pattern densities.

- **Toggle the Selection Style button.** This button turns the selection style ON and OFF. When OFF, the selected pixels are still selected but no red crosshatch pattern will be shown. When ON the red selection pattern will appear over the selected pixels. When making changes such as hue and brightness changes it is often quite obvious which pixels are selected by the changes being applied.

Try making a selection and then toggling the **Selection Style** button to turn off use of the selection pattern. The pixels are still selected but there is no pattern showing. We can then apply effects with a preview box checked and see the results directly within the image.
If we begin with a circular area of pixels selected we might wish to see the effects of lightness changes without visual interference from any selection style, so we toggle the Selection Style button.

At first we see nothing, since the selection style pattern is gone. However, the changing visual appearance on moving the lightness slider control in the Hue / Saturation effect makes it easy to see which pixels are affected.

**Selection in Drawings**

The examples above use images because it’s easy to see what is going on with selection in images. The same techniques will work throughout Manifold System with drawings, tables, charts and connections between them. For example, selections can be made and saved into the Selections pane for drawings as well as for images.

In almost all cases, every window in Manifold that shows data will have selection tools for mouse selection, the cumulative action of which will be controlled by selection modes.

Suppose we have a drawing of the United States. We can use Touch Select together with Select Add mode to click on states to add them to the selection…

First we click on California.
Next we click on Utah.

Arizona…

… and finally, Nevada. We can save this selection above as a saved selection in the Selections pane for the drawing.

Let's call this saved selection **South West States**

We can then make another selection in the drawing.
If we like, we can save this new selection in the Selections pane as well. We will call this new saved selection Mountain West.

At any time we can now click on either of the two saved selections in the Selections pane and, with the Preview button pushed in, see them previewed in the drawing.

For example, previewing the South West States saved selection will show it in the drawing in blue preview color.

**Saved Selections and the Transform Toolbar**

Saved selections will appear in transform toolbar boxes that can work with selection sets. For example, if a drawing has two saved selections called South West States and Mountain West then these will appear as choices in the transform toolbar when that drawing has the focus.

When drawings appear together in a map, if any drawing layer has the focus the transform toolbar for the map will list all saved selections in all drawings. This is a great convenience, but it also leads to a slight complication in that all saved selections using the same name are treated alike. Saved selections in the same drawing are required to have different names, but saved selections in different drawings might use the same name.

The transform toolbar for the map will treat all drawing saved selections using the same name as one, combined saved selection. For example, if we have two drawing layers in a map, one called East and the other called West and they both have a saved selection called Cities, then using the transform toolbar to create a convex hull using the saved selection choice called Cities will use the objects from both the East and West drawings that appear in their respective Cities saved selection.

Therefore, it is a good idea to use unique names for saved selections in drawings that might appear together in the same map. For example, we could name the saved selection in one drawing East - Cities and that in the other drawing West - Cities.

**Selection by Type**

The Edit menu for drawings includes a Select by Type choice. Use this command to select all of one or more types of objects, that is, all areas, lines or points in the drawing. The Select by Type command applies whatever Selection Objects and Selection Modes settings are in force.

For example, to select all of the points in a drawing we would push in the Select Points button and the Select Replace button and then choose Edit - Select by Type. All points in the drawing will be selected, replacing any previous selection.
If we simply wanted to add all the points to the current selection we would have pushed in the Select Add mode button and then chosen Edit - Select by Type.

CTRL-T is a keyboard shortcut to select by type. Other handy shortcuts are CTRL-A to select all and CTRL-I to invert the selection.

**Smart Mouse Selection**

Smart Mouse selection is used to select objects for editing. An object selected with smart mouse selection will appear with edit handles that can be used to move or reshape the object. An object chosen for editing is also called the primary selected object.

Commands involving the mouse are normally controlled by whatever toolbar button is engaged. To use Select Box for example, we push the Select Box button in and then use the mouse in this mode. To provide fast operations when the mouse is not engaged in any other command, Manifold allows “smart mouse” selection in a way that mimics ordinary Windows point and click mouse highlighting and motion methodology as is used in Word and other applications.

For example, CTRL-ALT clicking on an area object in a drawing will select it for editing. See the Editing Objects topic for examples and more information on interactive editing with smart mouse selection.

**Keyboard Modifiers with Smart Mouse Touch Selection**

Because a selection may already exist in the drawing when we wish to select an object for editing, smart mouse selection uses keyboard modifiers to allow a richer set of commands.

- **Click** Equivalent to Select Touch in Replace mode. Clicking into an empty part of the drawing deselects all objects.
- **CTRL-ALT Click** Select the object as the primary selected object, that is, select it for editing. Does not change the selection state of other objects.
- **CTRL Click** Invert the selection state of the object without changing the selection state of any other object. Equivalent to Invert mode.
- **SHIFT Click** Select the object if it is not yet selected. Does not change the selection state of other objects. Equivalent to Add mode.

**Smart Mouse Box Selection**

When clicked on an empty region and dragged the mouse automatically makes a box selection analogous to using the Select Box selection tool in Select Replace mode. This is a general shortcut for selection. To select an object as the primary selected object for editing, CTRL-ALT click on it.

**Keyboard Modifiers with Smart Mouse Box Selection**

The following keyboard modifiers may be used with smart mouse box selection:

- **Click and drag** Click on an empty region and drag to make a box selection in Select Replace mode. Selects all objects that are entirely
within the selection box.

CTRL Click on an empty region and drag to make a box selection in Invert Selection mode.

SHIFT Use an open box selection to select all objects any part of which are within the selection box.

Selection in Tables

When we open a table that’s linked to a drawing, the table window will show selected records with a light selection color background.

For example, the four states selected in the drawing at left will have their records in the table highlighted with selection color. (The records are alphabetical sort order so Nevada and Utah are not seen in the fragment of the table shown above).

The Selections pane shows the same saved selections for both a drawing and any tables associated with that drawing. If we choose a saved selection to be previewed, the records will be highlighted in preview color in the table as well as being shown with preview color in the drawing.

For example, if we click on the Table window to move the focus to the table, the Selections pane will still have two saved selections called South West States and Mountain West. We can click on Mountain West to highlight it and press in the Preview button to preview it.

As seen in the illustration above the states in the Mountain West saved selection will be highlighted in preview color in the table. Regardless of which window has the focus, a preview involving objects that appear in both drawings and tables will preview them in blue in all the drawing and table windows in which they appear.

It is clear from the hatch patterns which objects in the drawing appear both in the current selection and are being previewed from the Selections pane. In tables, records that are both selected and are in the preview will be shown in a color midway between the selection color and the preview color. By default, the combined color is light purple.

Selecting Records
Clicking on a record handle in the table is the same as a Select Touch on that record. We can also use CTRL-click and SHIFT-click.

In the illustration above we click on the record handle for the Nevada record to select it. Selecting a record will combine it with any existing selection in accordance with the selection mode (Replace, Add and so on) currently in force.

For example, we could change the selection mode to Add and then click on the record handle for the state of Alabama and several other states to add them to the selection shown in previous illustrations. The selection will be updated both in the drawing window and in the table window.

We can select items using any combination of selection in the map via mouse-based selection tools or selection in table windows via a variety of point-and-click highlighting methods within the table. For example, we could sort the table by clicking on a "population" field and then select only those records within a particular population range. We could then save this selection in the Selections pane. At any time this saved selection could then be combined with another selection made in the drawing or in the table.

This free-form way of selecting from within tables or maps is one of Manifold’s most original and most powerful concepts. Throughout Manifold we can get exactly the selection desired by clicking in different windows under the control of selection modes, saving selections within the Selections pane and then using selection command buttons from the Selections pane to combine saved selections with the existing selection. Sometimes it is very convenient to find records in tables by sorting columns and then selecting the topmost few records in the table to also select them in the map. See the Selection in Tables topic for more.

To select a range of records, click on the first record handle and then SHIFT-click on the last record handle. This will select all records (in accordance with the selection mode set) from first to last.

Note: clicking on a cell in a table makes that cell the active cell. To select a record in a table by clicking on a cell we can CTRL-click the cell. See the Working with Tables topic.

**Showing only the Selection in Tables**

To show only selected records in a table we open the table and push in the Filter Selected button either in the main toolbar or in the View - Filter dialog.

The table will then show only selected records. Records will appear and disappear from a table window filtered with Filter Selected as they are added to or subtracted from the selection by actions in other open windows.

**SHIFT key for "Open" Mouse Selection**
The keyboard’s **SHIFT** key may be used in combination with mouse selection to “open” the mouse selection.

By default, mouse selection commands will select all those items that are **entirely within** the selection shape that’s drawn. The **SHIFT** key “opens” the selection so that it grabs all items any part of which is within the shape drawn with the mouse. For pixels in images, or points in drawings “open” mode is the same as the default. For lines or areas in drawings “open” mode is a way of selecting all objects that touch or are within the mouse box.

Drawing the above box over the US_Main map selects only those states that are entirely within the box.

In this case, just one state.

If we hold down the **SHIFT** key while making the same selection…

…we select all states that touch or are within the box.

**SHIFT** Touch Selection in Images
Just like using the **SHIFT** key causes a more inclusive selection in drawings, there is also an analogous **SHIFT** key function for touch selection in images.

Using touch selection to click on a white pixel in the image at the left will select all pixels within the tolerance threshold that are contiguous (i.e., adjacent to each other). In this case, white pixels between the legs are selected but not those on other sides of the legs.

Using **SHIFT - touch** selection to click at the same spot will select all pixels within the tolerance threshold throughout the entire image no matter where they are located. In this case, all of the white pixels within the tolerance level have been selected throughout the image.

**Tip:** Using **SHIFT - touch** selection is a good way to select pixels to replace a color throughout the entire image. Select the color throughout the entire image and then use either Hue / Saturation to change the color or the Paint Bucket tool with the **SHIFT** key to pour a different color into the selection throughout the entire image.

### Speed of Select Touch in Images

Determining whether or not pixels are within a contiguously connect region requires many spatial computations, so **SHIFT-touch** selection frequently will operate much faster than touch selection.

### CTRL Key and Select Touch in Images

Holding down the **CTRL** key while using Touch Select with images will select only that pixel that is touched. **CTRL - touch** is used when zoomed far into a map to “clean up” selections pixel by pixel.

For example, the illustrations above show an image at great zoom where we wished to select only the lighter regions. The image at left shows that one of the black pixels is selected. We can deselect it by clicking it using **CTRL-touch** with selection mode set to **Subtract**. This method allows us to easily add or subtract individual pixels from the selection.

### Autoscroll and Selection

When making freeform selections, Manifold will autoscroll the window to allow us to move the window even in the middle of a selection command. Autoscroll may be turned off or configured in Tools - Options.
**Touch Selection Accuracy**

Touch selection will select objects and pixels within a 4 screen-pixel radius of the actual screen pixel touched. This setting makes it possible to "touch" objects reliably without frustration even on higher resolution monitors. If touch selection chooses unwanted objects or pixels that are very close to the location touched, zoom in to provide greater spacing and touch again. At greater and greater zoom levels, individual pixels within images will enlarge until they take up many screen pixels as seen in the illustrations above.

**Touch Selection Tolerance Setting**

Touch selection will select all pixels whose color is within a given tolerance range of the pixel touched by the mouse. For RGB or RGBA images, tolerance is computed only on those channels that are made visible in the layers pane. For palette images, tolerance is computed based on the R, G and B values of the palette color as compared to other palette colors. Change the tolerance setting as desired in the Tool Properties pane to increase or restrict the range of pixel color values that are selected when using touch selection.

In the illustrations above we have increased tolerance from a low level to a much higher level and then clicked on the blue sky just above the head of the monument. As tolerance is increased a greater number of pixels are selected in colors that are more and more different from the blue hue of the touched pixel. At high tolerance settings pixels with colors very different from the touched pixel will be selected.

**Selections are Shown in their Map Layer**

Selections are shown in all of the image or drawing layers in which they occur. If a selection is made only in a lower layer, upper layers might cover it.
Suppose we have one RGBA image based on our sample *bronze* image that consists of regions of visible, partially transparent pixels as well as invisible pixels. We can use **Select Circle** in the image window to select only that part of the circle that is over visible pixels. It selects all visible pixels, even those with low opacity specified using RGBA pixel transparency.

Suppose we open another image window showing our standard *schloss* image. We can make a **Select Box** rectangular selection in the schloss image window.

We can display both images together in a map window. The selections made in each image window will appear in the layer associated with their image. Seeing the two selections together in the map window shows both selections have been made independently.
Suppose now we click on the bronze image window and deselect the selection made in the bronze image and change the Selection Style to use a border style. The illustration above shows the resultant map window view. Because the bronze image is above the schloss image in the map layer stack, the head of the bronze image covers not only the schloss image pixels but also the selection indication. Since the bronze image is an RGBA image that is partially transparent even near the head, if we look carefully we can see a hint of red selection border peeking through the bronze image.

Note that selections are properties of the individual components, not of a map overall. A "map" is just a stack of layers of components like images and drawings. Clicking on a layer's tab to make it active directs the focus of what we do to that layer. Any selections we make by default will be for that layer.

The above examples clicked open image windows for the bronze and the schloss image so that selections could be made in those image windows. Selections can also be made in the map window, in which case they apply to the active layer if it is visible. By default, using a selection tool in an active layer if that layer is not visible has no effect. Selection tools apply to all visible layers when the ALT key is pressed even if the active layer is not visible. Had we made the rectangular selection in the map while pressing the ALT key, we would have selected those pixels both in the bronze image as well as in the schloss image that fell within the selection box.

Selection in Maps / ALT Key in Maps

Selection and Clipboard (Cut, Copy, etc) operations in maps work on the active layer by default. To extend the action of these operations to all visible layers, press the ALT key. For example, drawing a selection box in a map will only select objects in the active layer. Holding the ALT key while drawing a selection box will select objects from all visible layers. Pressing Delete deletes selected objects in the active layer only. Pressing ALT-Delete deletes selected objects in all visible layers. Think of ALT as a mnemonic for "All."

Menu items that appear when a map window has the focus will apply only to the active layer. For example, Edit - Select All, Select None and Select Inverse apply to the active layer only, as does Edit - Copy. Menu commands cannot be extended to include all layers. The ALT key works only with mouse selection methods and keyboard commands. It does not modify menu commands (such as Edit - Copy) which always work on the active layer.

To apply selection commands to all visible layers, use the ALT key together with a mouse-based selection command such as Select Box or with a keyboard short cut such as CTRL- A or with a keyboard shortcut such as CTRL-A. For example, and ALT-Select Box used with Invert selection mode will create an inverse selection over all visible layers. ALT-CTRL-A will select all objects from all visible layers. If all objects in all layers are selected, ALT-CTRL-I ("select inverse" applied to all objects) will deselect all objects.

See the Layers and Commands topic for an example of the above.

**Image and Surface Selection Commands**
Certain selection commands are available only in images and surfaces. See the Selection using Masks and Modifying Selections topics.

**Menu Selection Commands**

Don’t forget about the selection commands available under the *Edit* menu (See also the Edit - Select All / None / Inverse topic for handy menu based selection commands):

- **Select All**: Select all objects or pixels in the component.
- **Select None**: Deselect all objects or pixels in the component.
- **Select Inverse**: Invert the selection. Those items that were not selected will become selected, while those that were selected will be deselected.

*Select All* and *Select None* are obvious in action, as seen above using an image as an example.

*Select Inverse* is a very useful command, because sometimes when making a selection it is easier to pick out those things we do not want to select. For example, if we ultimately would like to select the land area of South America in the images above, it is easier to first use *Touch Select* to select ocean areas and to then use *Edit - Select Inverse* to get the land areas. The ocean regions are very even in color and can be selected with a single click. The land areas contain many different colors and would take many clicks when using a reasonable tolerance setting with *Touch Select.*
These commands also work with drawing layers. In the sequence above, we would like to retain a circular region of lines in a drawing layer while deleting all other lines. We do this by selecting the circular region, choosing Select Inverse and then deleting the new selection (all the lines except the ones we want).

These commands also work with tables and even forms.

For example, if one control in a form is selected, using Select Inverse will select the other controls.

**Transferring a Selection between Components**

The Transfer Selection command can transfer a selection from one component to another when the two components are used together as layers in a map. For example, we can use it to select all pixels in an image that are within selected areas in a drawing. See the Transfer Selection topic for more.

**Keyboard Zoom Commands during Selection and Editing**

When making selections or editing objects (such as inserting areas using the Autocomplete feature) we will often want to zoom in or out in the middle of a selection or editing command. Use the + and - keys on the keyboard to do so. Usually the + and - keys in the numeric keypad are used.

- **“Plus” key:** Zoom in at the current tool position.
- **“Minus” key:** Zoom out at the current tool position.

The current tool position is the position of the mouse at the time the + or - key is pressed. When editing, using these keys allows us to easily zoom into a desired location, make a few precise clicks and then zoom back out and proceed at the usual scale.

**Status Bar Selection Readout**

The leftmost portion of the status bar has multiple uses. It normally is used to show extended messages that accompany tool tips when the mouse cursor hovers over a command button.

2135.9333 x 1996.6333 m

For some commands, like selection commands, the leftmost portion of the status bar will be used to show the dimensions of the area selected in the units used for the current projection taking into account the local scale and units settings in the Edit - Assign Projection dialog for that component. If the component is georegistered the sizes shown will be true dimensions.

**See Also**

ViewBots for an example using ViewBots to make selections using Selection Commands from the ViewBots pane.

Query Toolbar for selection via the query toolbar.
Selecting Objects with Queries
Transform Operators - Drawings for a list of transform operators that make selections.

Advanced
See the Dialog Mode and Visual Tools topic for a parameter or value oriented way of specifying a selection shape.
Smart Mouse Selection

"Smart Mouse" selection is a shortcut, simplified way to make selections that is similar to selection methods used in editing operations with many Windows applications. Smart mouse selection ignores selection modes and selection object filters and works only if the mouse is not busy in a command mode. As such, it is particularly useful when editing controls on forms or for casual editing of drawings, when the mouse is not usually engaged in a command. Visual Studio users will recognize the similarity between smart mouse selection with Manifold forms and forms editing in Visual Studio.

Pressing the ESC key will escape from any tool mode, such as zoom tool or selection tool and will restore default smart mouse selection. Pressing ESC is therefore a handy way of quickly exiting from some tool mode and getting back into default smart mouse selection.

In Manifold, commands involving the mouse are normally controlled by whatever toolbar button is engaged. To use Select Box for example, we push the Select Box button in and then use the mouse in this command. If the mouse is engaged in a different command mode (such as Zoom In or Zoom Out) the mouse works within whatever command has been chosen. Sometimes we would like fast and simple selection when the mouse is not engaged in any other command. For example, when moving objects around in maps we often want to select only one object and then drag it to another position. In such cases it is faster to do selection in a "modeless" way. Smart mouse selection provides a fast, simple and modeless way of doing selection in such circumstances.

Smart mouse selection is used to select items for editing in drawings, forms, labels, and layouts. An object selected with a CTRL-ALT click will appear with edit handles that can be used to move or reshape the object. An object chosen for editing is also called the primary selected object.

This topic is written using drawings and drawing objects as examples. The same moves will work with items in layouts, labels or controls in forms as well. See the Editing Forms topic for examples of using smart mouse selection with forms.

Smart mouse selection is intended for fast, casual selections during editing operations. Smart mouse selection does not use selection modes. To make selections using selection modes, use the selection tools described in the Selection topic.

Smart Mouse Touch Selection

The simplest form of smart mouse selection is to click on an object to select it. This is a smart mouse version of Select Touch. For example, clicking on an area object in a drawing will select it. CTRL-ALT-clicking on an object will select it for editing, as seen above.

To deselect an object from editing, CTRL-click on it. This toggles it to unselected mode.

See the Editing Objects topic for examples and more information on interactive editing with smart mouse selection.

Keyboard Modifiers with Smart Mouse Touch Selection

Because a selection may already exist in the drawing when we wish to select an object for editing, smart mouse selection uses keyboard modifiers to allow a richer set of commands.

Click   Equivalent to Select Touch in Replace mode.

Clicking into an empty part of the drawing deselects all objects.
CTRL-ALT Click  Select an object for editing, making it the primary selected object. Does not change the selection state of other objects.

CTRL Click  Invert the selection state of the object without changing the selection state of any other object. Equivalent to Invert mode.

SHIFT Click  Select the object if it is not yet selected. Does not change the selection state of other objects. Equivalent to Add mode.

Smart Mouse Box Selection

When clicked on an empty region and dragged the mouse automatically makes a box selection analogous to using the Select Box selection tool in Select Replace mode. This is a general shortcut for selection. To select an object as the primary selected object for editing, CTRL-ALT click on it.

Keyboard Modifiers with Smart Mouse Box Selection

The following keyboard modifiers may be used with smart mouse box selection:

- **Click and drag** Click on an empty region and drag to make a box selection in Select Replace mode. Selects all objects that are entirely within the selection box.
- **CTRL** Click on an empty region and drag to make a box selection in Invert Selection mode.
- **SHIFT** Use an open box selection to select all objects any part of which are within the selection box.

Use within Layouts Components and Forms

Although we can always use the regular, dedicated selection commands covered in the Selection topic, for some components smart mouse selection is the most typical way of making selections. For example, smart mouse selection will be used by most users as their primary means of making selections.

Notes

Why is it called "smart mouse" selection? This is based on the idea that to support this type of selection the mouse cursor in its default mode must understand a lot about its environment. For example, it must be able to keep track of and know when it is clicked on a drawing object (and, for that matter, when the object is in the active layer if the mouse hovers in a map window) and much more. This requires considerably more thought for the cursor than merely being a default Windows cursor, so it is said to be a "smart mouse" cursor.

Note that the default action of the mouse in "smart mouse" moves depends on the context of where it is clicked and how it is moved. For example, clicking on an empty region and dragging creates a box selection mouse move. Clicking on an object and dragging is a drag-n-drop operation.

Inconsistency in Selection Methods

Advanced Manifold users will notice that there is a slight inconsistency in how selection operates in different settings, in particular how selection works in drawings when using smart mouse mode and how selection works in tables.

The general idea in Manifold is to have selection governed by both the selection mode desired (replace, add, subtract, invert, intersect) as well as by the operation of whatever selection tool has been put into play (Select touch, Select box, etc.). This general idea is modified in two cases, smart mouse selection in drawings and selection in tables, to allow simplified operation in smart mouse selection and to conform to user habits fostered by other Windows software in the case of tables.

Smart mouse mode uses neither selection tools nor selection modes. It provides a simplified, default way of selecting objects. In drawings, if a selection tool such as Select touch or Select box is used the selection mode
is always honored. Default smart mouse mode, which is available when some other tool is not in play, does not use the current selection mode. The advantage of such a difference between regular selection tools and smart mouse mode is that smart mouse mode can be a simple, default way of selecting objects. The disadvantage of this difference is that it is a difference and thus something that must be remembered.

Selection in tables will honor selection modes, but the visual selection tools used with drawings are not used in tables, as we can see by jumping ahead to the Selection in Tables topic. Because operations with tables are common in many Windows applications, in the case of tables Manifold adheres to the common Windows practice of selecting records in a table by clicking on a dedicated "handle" column. Clicking on a record handle in a table window always uses the current selection mode.

See Also

Selection for selection using selection modes, selection objects and the full range of selection tools in Manifold.
Editing Forms for examples of smart mouse selection with forms.
Layouts for a brief example of smart mouse selection within print layouts.
Editing Objects for usage of smart mouse selection when changing the shapes of objects.
Formatting

Formatting is the application of style, color and size to objects in drawings or themes or to text in labels components. Because drawings usually appear as layers in maps, we will often think of formatting as formatting layers in maps. We use the Format toolbar to specify formatting that is to be used with objects in drawings and labels in text layers.

Formatting is not necessary in images because the appearance of images is controlled exclusively by the colors of the pixels that comprise the image. The Format toolbar is used in images to set the color and style applied by various painting tools.

A Familiar Concept

Formatting in a GIS context is directly analogous to formatting in a word processor such as Microsoft Word. When we format text in a word processor we apply display attributes such as font style, font size and color to the text.

The quick brown fox

The quick brown fox

Changing the font size, style and color can result in dramatic changes in the appearance of the text but the actual words stay the same. Just so, changing the formatting of objects in drawings can change dramatically their appearance even though the actual objects within the drawing stay the same.

The first illustration above shows default formatting for areas, lines and points. The second illustration shows those same objects with garish formatting applied as seen in a map with multiple layers (each layer having a different format). For example, one point is shown using a point style showing a yellow bus instead of the default small round circle style. Note that the lower horizontal line really consists of three lines, one of which was formatted as a thick red line while the other two were formatted using a dotted line style. In addition, layer opacity was used for the layer containing the checkerboard triangle.

Changing the format does not change the objects in the map. Using a bus symbol to show the point doesn’t turn the point into a large, bus-shaped area object. It simply uses a different icon instead of a small round circle to show where the point is located.

Formatting Requires Some Effort

Drawings will often be imported into Manifold from file formats that do not retain any information about the desired display attributes of the objects. This is analogous to importing a document into Microsoft Word from a plain .txt file. In Word, the entire document will initially appear in Courier New or some other default font. Depending on the source of the document it might not have margins or paragraph marks correctly set and so on. Converting
such a plain, .txt file into a beautifully formatted Microsoft Word .doc file will require some effort and good taste in the choice of fonts and other formatting details.

In Manifold System, we will often import complex maps from formats that do not retain information about visual appearance. In such cases, Manifold will initially show the drawing using default formatting. Specifying formatting to create a pretty map from such raw drawings will require some work and good taste.

Both illustrations above show the same map that consists of several drawings: the illustration on left uses default formatting for the drawings. The illustration at right has had formatting applied to the drawings so that land areas are colored green and ocean areas blue with population points shown as yellow icons instead of small round circles. Note how the application of sensible formatting makes the image on the right more comprehensible.

Manifold has many tools to help format complex maps with great efficiency. Also, many basic maps are provided on the Manifold downloads site have already been formatted. These maps may be used “as is” or may be easily reformatted to suite any taste.

The Formatting Drawings topic covers formatting in detail. It should be read together with the other drawings topics, beginning with the main Drawings topic.

**Thematic Formatting**

Thematic formatting is the process of automatically coloring objects in drawings or themes based on the value of their data attribute fields in tables associated with the drawing. For example, we might change the color of states by the value of their populations or increase or decrease the size of points representing cities based on the value of their population fields. The greater the population, the larger the point.

The illustration above shows cities in Texas where each city has a population field in the associated table. The point sizes of the cities have been thematically formatted by the value of the population field. The above illustration also has had the background color of points thematically formatted by the population field, so that the color changes as well.

See the Thematic Formatting topic for details on thematic formatting.

**Customization**

Palettes used in thematic formats may be customized, and new palettes may be added to Manifold. See the Customization topic.
New point styles may be added to Manifold by including True Type font symbols or images in .bmp, .jpg or other graphics files as point styles. For example, the bus style used in the illustration above comes from a popular Windows font. See the Custom Point Styles topic for more.

**Themes and Drawings**

It’s frequently the case that we would like to format the same drawing in several different ways. Manifold allows us to create **themes** to show the same drawing using different formatting characteristics. Themes require no extra storage because they do not duplicate the drawing, they simply show the same drawing using different formatting, such as different background or foreground colors. See the Themes topic for details.
Adding Text Labels

Text labels are added using **Labels** components. A labels component is like a drawing that contains only text items called **labels**.

There are two different types of labels components, **bound** and **unbound**, for which text labels are created in two different ways.

- **An unbound** labels component begins as a blank labels component that contains no labels. To create labels within an unbound labels component, we must use the **Insert Label** tools from the tools toolbar for labels to manually click at each spot where we want a label and enter the label text.

- **In a bound** labels component the positions of labels and their text can be automatically taken from objects in a **parent** drawing. Creating a bound labels component automatically creates all the labels in it as well. For example, suppose we want to automatically create text labels for every province in a drawing of a country. We could create a labels component bound to that drawing where the name of each province is taken from a "**Place Name**" field in the drawing's table. When the labels component is created, it will have a label for each province centered on the province that takes its text from the **Place Name** field. If a new province were added to the parent drawing, a new label would be created for it automatically in the labels component, since a bound labels component is always automatically synchronized to its parent drawing.

There are two types of labels components because sometimes we would like labels to be automatically created and sometimes we would like to create labels manually. For example, when using Manifold's **Internet Map Server** capability we don't know what view or zoom level visitors to the web site might select, so it would be useful in this case if Manifold would automatically create and place labels for us. On the other hand, we may spend considerable effort manually creating and positioning labels on a map for printed publication where there is only one view and zoom level.

At times we may want to combine both automatic and manual methods. For example, we might want to create labels at manually-specified positions but have Manifold populate the label text automatically from a data attribute. As will be seen in the detailed labels topics in this documentation, that is possible as well.

A labels component that creates bound labels is said to be **bound** to the drawing the table of which it uses. In the project pane it will be shown indented under the drawing at the same level as the drawing's table. An unbound labels component is not related to any drawing and is shown as a standalone component in the project pane hierarchy.

![Unbound Labels Component](image)

Labels within an unbound labels component appear as text within an otherwise transparent window, as seen above. Just like in drawings, the Layers pane can be used to turn the background on and off. When working with unbound labels components, it usually makes sense to work with the labels component as a layer in a map where other layers, such as a drawing, can provide some visual context for the position of the labels.
Labels that are bound to a drawing will be displayed with the parent drawing shown in a grayed-out schematic view in the background. This provides a visual context for the labels.

**Creating Labels Manually**

Labels can be created manually only within an unbound labels component. Because it is difficult to see the exact placement of labels without using a drawing or image as a guide, manually created labels are almost always created using a map where the labels layer is above the drawing or image layer.

**Adding Labels to a Map Manually**

1. Create a map using the desired drawing.
2. Right click on a layer tab and choose **Add - New Labels**
3. In the **Create Labels** dialog that pops up, do **not** choose any components in the **Parent** pane. Click **OK**.
4. Click the **Insert Label** button. In the labels component, click wherever a label is desired. In the popup **Text** dialog, enter the text for the label and press **OK**. [Pressing **Enter** starts a new line, since Manifold labels can be multi-line labels.]
5. When all desired labels have been entered, click out the **Insert Label** button.
6. Format labels in the labels layer as desired using the Format toolbar.
7. If you would like to have different styles or colors of labels, repeat the above process with another labels layer to create new labels in that layer. Each layer can have a different format. Alternately, if the appearance of labels is to be controlled by a data field in the drawing, use thematic formatting to specify the size, style, foreground color, background color or rotation of the labels based upon the field.

The **Insert Label** command is used to insert a label. Click to indicate the spot where the label is to appear and then provide the desired label text within the dialog window that appears. Click **OK** to accept the text. Note that labels can have multiple lines, so an **Enter** key starts a new line.

The Format toolbar is used to specify the text formatting (font, font size, etc.) throughout the entire layers component, although individual labels may be formatted differently by selecting the label first and changing its format.
Changes are immediately applied.

Labels components are usually used within maps so that other layers can provide a visual context for the location and scale of the labels. We can stack up as many labels components within a map as we like, using a different format for each labels layer to utilize many different text label formats within the map.

Manifold can also use the value of a field to specify the size, style, foreground color, background color or rotation of the labels. See the Thematic Formatting and Labels topic for details.

Creating Labels from Fields

By using bound labels Manifold can automatically create labels using information from fields. Suppose we have a drawing of Europe showing each country as an area, for example, and for each area we have data fields such as the country's name, population and other fields. We can create a labels component that takes the text for its labels for each country from one or more desired fields. See the Creating Labels from Fields topic for detailed instructions.

See Also

Labels have many more options than are discussed in this brief introductory topic. See the main Labels topic and other topics in the Labels chapter for more on labels. For example, see the Aligning Labels topic for alignment options.

Note

In addition to using Add - New Labels from map's layer tab context menu, we can create new labels components at any time using File - Create - Labels. Because maps (even maps created with default settings) use geographic projections, beginners are strongly advised have the focus on the map window when creating labels components that will be used in the map. This guarantees that the labels component will be automatically created within a geographic projection that matches the map.

Tech Tip

By default, Manifold will autoclip labels (not show some labels) to prevent the display from being too crowded with labels. If you add labels and they disappear or some other labels disappear, they are being autoclipped by the overlap resolution algorithm. To turn off autoclipping open the labels component in its own window, choose View - Display Options and uncheck the Resolve overlaps box.
Adding Legends

A Legend is a small display that provides captions, summarizes formats or otherwise explains what is seen in a window or in a print layout. Legends appear to float over the window and are always on top of other items in the window.

Legends may use simple, default styles or they may consist of complex and sophisticated items. Legends will often show object formats for drawings, surface colors by height for surfaces, thematic formats for labels (if one is present), and colors in images.

Although legends are part of a component like a drawing, they are an accessory item like a north arrow or scale bar associated with that component and are not the same as the actual contents of the component like the points, lines and areas that make up a drawing. Legends are not edited using tools such as painting tools for an image or the various editing tools available for drawings. Instead, legends are edited using settings in the Legend dialog.

Except for layouts, each component has one legend. A component remembers the settings for its legend as part of the properties of that component. Print layouts may contain multiple legends, including even multiple copies of the legend for each component that appears as an element in the layout.

An entire legend or any part of a legend may be composed for us automatically by the system or we can manually specify all or any part of a legend. Automatically-created legend items may be dynamically updated if the information they show, such as thematic formatting, changes. Dynamic and static parts of legends may be mixed, so that part of a legend (such as a title) may be manually specified while other parts, such as the colors and values for thematic formatting, may be dynamically generated by the system.

To add a legend:

1. Open the component.
2. Choose View - Legend to open the Legend dialog.
3. Check the Show Legend box and change the settings as desired. Press OK.

The default alignment is to place the legend in the right bottom corner of the window. If the window is resized or the view panned or zoomed the legend will stay in place in the right bottom corner of the window. To place the legend in any position, choose None in the Align box of the Legend dialog. The legend may then be dragged with the mouse to any desired position in the window, where it will stay no matter how the drawing or map is panned or zoomed.

By default, legends will be created showing elements for all types of objects using a simple style. Legends for maps will show elements for all components in the map.

To change a legend:

1. Double click the legend, or right click on the legend and choose Properties, or choose View - Legend.
2. Adjust the settings in the Legend dialog as desired and press OK.

In addition to Properties, the context menu that appears when right clicking on a legend also allows us to Hide the legend or to choose Align for different alignment.

Legend Styles
Manifold legends use a simple style that provides two main options:

- **Caption or no Caption** - Entering text into the Caption box adds a main caption, that is, title, to the top of the legend using inverted foreground / background colors to set it apart. When caption text is added Manifold will automatically add an additional panel to the top of the legend.

- **Standard style or Transparent** - Using a transparent style replaces the usual background color with no color.

Except for the above two adjustments to overall style, legends use a single style that consists of a series of elements, which are one or more lines of items in the legend. Although elements are by default stacked vertically...

...the clever use of vertical separator elements can force legends to have a more horizontal appearance, as seen below.

Within the legend the appearance of each item will be controlled by first, the choice of what type of element is used and second, by the various options specified for that type of element. Different types of elements have different options, in some cases rather extensive and elaborate options.

For example, text elements can have their fonts, size and attributes (such as boldfacing) controlled, while elements that show a sample of area formatting can be either manually controlled or automatically derived from the thematic formatting of a drawing. There are many different types of elements that may appear in a Manifold legend as well as many different display options within various types of elements.

**Example**

The following examples show different combinations of use of caption together with use of standard or transparent style. To the right of each legend is a screenshot of the Legend dialog used to create it (only the relevant part of the Legend dialog is shown).
The default setting is no text in the Caption box so no caption appears. The default setting is Standard in the Style box so the legend appears as a rectangle of background color within a border of foreground color.

If we enter some text into the Caption box the caption appears in an inversely-colored section of the legend.

If we switch to Transparent in the Style box the background color and surrounding box disappear.

Removing text from the Caption box removes the caption portion of the legend. Although there are many cases when transparent style is useful, as can be seen from this example it is probably not a good idea to use transparent style for legends that consist of fine text appearing against a busy background.

Example

We will add a simple legend to the Mexico sample drawing and then modify it.
We open the Mexico drawing in a drawing window. To add a legend we choose View - Legend from the main menu.

In the Legend dialog we check the Show legend box. The dialog appears with default legend elements. Thinking ahead a bit, we change the Align box to None so that later we will be able to move the legend about as desired by dragging it to whatever position we want in the drawing window.

Press OK. (To save space, the entire dialog is not shown in the illustrations for this example.)
A legend with no alignment specified initially appears in the upper left corner of the drawing window.

We can move it in an instant to a new position by clicking and dragging it to whatever location we desire.

This assumes, of course, that the mouse is not occupied by being in the middle of some other command.

To change the legend, we can choose View - Legend again from the main menu or we could right-click on it and choose Properties to launch the Legend dialog...
...or we can simply double click on the legend as a shortcut to launching the Legend dialog.

Within the Legend dialog we can check the Customize Legend box to enable editing of legend elements. With this box checked we can now select individual legend elements and change them as we see fit.

For example, we can click on the Mexico Drawing text element at the top of the dialog to select it for editing. The colored background bar shows which element is selected for editing.
Pressing the **Delete Element** button in the toolbar will delete this element.

This simplifies the legend to only one element, an element that shows an area sample that has **Areas** as the text.

The text for this item is generated using an **escape sequence**, an expression in [] brackets for which the system automatically generates the relevant text. In this case, the default text is element is the escape sequence **[Type / Column]** which the system will fill in with the type of object, **Areas** in this case, as well as the column used for formatting if thematic formatting is in play. Since this drawing does not use thematic formatting there is no mention of a column in the automatically generated legend text.

Suppose we would rather have the text be **Provinces** - we can make that change by clicking into the **Text** box to change the text.
We enter the text **Provinces** and then we press **OK**.

The result is a simplified legend positioned where we want containing the text we want.

The art of creating legends in Manifold is simply the above procedure, at times greatly elaborated to create whatever legend we want. We learn to use the **Legend** dialog to add and delete elements to legends as desired. We can have some elements created for us by Manifold or we can add elements manually using a very wide range of options or we can combine automatically-created elements with manually-added elements in the same legend. Different types of components, such as images or drawings, will have different options enabled in the **Legend** dialog. Individual types of elements will also have different options available, such as a range of display styles for color wells or a wide range of font controls for text elements.

We can pick and choose from different legends by copying and pasting elements between legends. We can have Manifold automatically assemble legends for complex components like maps and then flatten those legend elements into individual, non-automatic elements we can customize as we like. We can use escape sequences to insert text that the system automatically generates for us in a wide range of circumstances. The result of combining all of these options is the ability to create legends of great flexibility while retaining the convenience of having the system create simple legends for us automatically.

**See Also**

**Legends**

**Legend Text Escape Sequences**
Editing

Manifold includes a wide variety of methods for creating and editing images, drawings, tables and other components. Some methods, such as standard Windows Clipboard methods will work in almost all windows, while other editing methods are specific to the type of data (drawing, image, table, etc) being edited. Many editing operations (Cut or Copy, for example) will begin with making a selection. Other editing operations will automatically work on just the selection if one is present and otherwise will work on the entire layer or table.

Copy and Paste

Virtually anything within Manifold can be moved about using Windows Clipboard commands such as Edit - Copy and Edit - Paste. When used in drawings and images, such commands apply to objects and pixels that have been selected.

Interactive Editing

Interactive editing uses a different collection of toolbar buttons depending on whether we are editing drawings, images or other components. For example, we use the Tools toolbar to draw lines or areas in drawings. See the Editing Drawings topic for more.

Different components will also have different collections of editing instruments. For example, the Tools toolbar is not enabled for tables.

Smart Mouse Editing

When the mouse cursor is not occupied with a command mode, default "smart mouse" selection and editing applies. See the Editing Drawings topic.

Transform toolbar

The Transform toolbar is used to make global changes that potentially apply to many items at once. The user interface and usage style is the same for all components for which the Transform toolbar is enabled, but the specific functions within the toolbar will change to suit the type of component involved.

We use the transform toolbar for many routine commands, such as cropping an image. When editing drawings or images it is one of the most frequently used controls in Manifold.

Image, Drawing and Other Menu Commands

Each component has menu commands available that can alter the component. See Editing Drawings, Editing Images, Tables, Editing Surfaces and similar topics. There are literally hundreds of editing capabilities within nearly all parts of Manifold.

SQL

SQL queries may be used not only to show data or select it in different ways, they can also be used to update tables by changing the data they contain.

Scripts

We can write programs using Visual Basic scripting or JavaScript that can alter objects, recolor pixels in images and otherwise edit components.

Moving Objects by Editing Tables

To move points or other objects by editing their coordinates in tables, make sure to read the Editing Data in Tables topic, the Intrinsic Fields topic and the Editing Intrinsic Fields in Tables topic.
We can add, delete or move points or lines in Linked Drawings by editing the geocoded tables used to create the linked drawing.

**The Object Coordinates Dialog**

Right clicking on an object and choosing Coordinates launches the Object Coordinates dialog to allow direct editing of the coordinates that comprise the object. Right clicking onto an edit handle and choosing Coordinates will launch the Object Coordinates dialog with that particular edit handle coordinate highlighted.

**The Object Fields Dialog**

Double-clicking an object in a drawing launches the Object Fields dialog that shows any table fields associated with this object. Fields may be edited within the Object Fields dialog.

**The Info Pane**

An object shown in the Info Pane can have its fields edited.

**Notes**

By default Manifold uses progressive rendering to fill in the contents of display windows, allowing users to continue work while redisplay occurs in windows that show large data that might take more than a few seconds to render. Lengthy operations such as transforms automatically suspend progressive rendering and resume it upon completion, so that the system does not lose time rendering what might be obsolete data.

**See Also**

**Editing Objects**
Dialog Mode and Visual Tools

Manifold provides a wide range of mouse-based, visual tools. These include various tools used for editing and selection in drawings and images, painting tools in images as well as other commands that work with the mouse such as the tracker tool used to measure distance, the grabber panning tool and others.

While it is very useful to be able to visually click and drag and otherwise choose the range of action of tools using the mouse, sometimes we need to guide the operation of such tools using exact parameters. For example, we might want to draw a line from one exact location to a second exact location where we would like to specify the locations using numeric parameters such as latitude and longitude locations and not try to approximate the locations using mouse motions. Or, as is often done in COGO applications, we might want to draw a line from a starting location to another location where we would like to enter numeric values for the bearing and distance of the line to the new location.

**Dialog mode** allows us to work with a wide variety of commands using a dialog interface to enter specific values. We launch the visual tool desired, and then click at least once to start the tool and then while the tool is still in action press the **Insert** key on our keyboard to launch dialog mode. While in dialog mode the action is controlled by the values we enter into dialog's parameter boxes and the dialog commands we use.

With many mouse-based commands we can go back and forth between visual, mouse mode and dialog mode to take advantage of dialog assistance when we want it and otherwise continue using mouse tools in the usual visual way.

Because dialog mode uses the actual coordinates employed by whatever component we are working with, for most purposes it requires a clear understanding by the user of what coordinate system (projection) is in use, what units it uses and how coordinates must be specified in that projection to exactly identify a location.

Depending on the tool with which we use dialog mode and our objective we will probably use only part of the dialog while ignoring other parts, and we may want to combine the use of visual mouse tools, dialog mode and other Manifold features such as snap mode for the easiest and most effective work flow.

**Note:** This topic uses concepts that beginners may find challenging. New users will find it profitable to read this topic and then to read it a second time after practical experience with editing and other techniques. Nonetheless, this topic appears in the **Introduction** because dialog mode pops up so often (pun intended) as a very useful method in many Manifold operations.

**Ellipses and Circles**

Pressing **Insert** when using a visual ellipse or circle tool when editing, painting or selecting will pop up the **Ellipse** dialog. The dialog allows entering the coordinates of the center of the ellipse or circle and the height and width dimension. The **Ellipse** dialog is used for both ellipses and circles since a circle is just a special case of an ellipse that is as tall as it is wide.

We operate the dialog by entering the parameters we would like and pressing **Apply** or by pressing **Continue** to return to visual mouse mode.

We can specify the ellipse of action either by providing the left, right, upper and lower bounds and pressing **Apply** or by providing the location of the center and the width and the height and pressing **Apply**.
Bounds  Use this section to specify the ellipse of action by providing the coordinates of the bounding box of the ellipse.

L  Left - The X coordinate of the left edge of the ellipse.
R  Right - The X coordinate of the right edge of the ellipse.
T  Top - The Y coordinate of the top edge of the ellipse.
B  Bottom - The Y coordinate of the bottom edge of the ellipse.

Center / size  Use this section to specify the ellipse of action by providing the coordinates of the center of the ellipse and its width and height.

X  The X coordinate of the center of the ellipse.
Y  The Y coordinate of the center of the ellipse.
W  Width - The width of the ellipse in units employed by the coordinate system of the component.
H  Height - The height of the ellipse in units employed by the coordinate system of the component.

Apply  Apply either the Bounds settings or the Center / size settings to specify the ellipse in dialog mode.

Continue  Abandon this dialog and return to visual mode.

Parameter boxes in the dialog will be loaded with whatever coordinates are implied by the position of the mouse at the time the Insert key was pressed to launch dialog mode. For example, an "insert on center" form of an insert circle command for drawings will have the location of the central point clicked already loaded into the X and Y boxes.

Example

We will use dialog mode to create a circular area exactly 1500 meters in radius from a given point.
We begin with a map that has three drawing layers: a layer with roads, a layer with points of interest shown with green dots and a third layer in which we will draw a circular area. The third area has already been formatted with partial opacity so we will be able to draw areas and still see the roads “through” those areas.

The drawings and map all use Orthographic projection, which uses a coordinate system defined in meters.

To make it easier to click exactly on points, we will turn on Snap to Points mode.

Next, we will use the Insert Circle on Center tool to create a circle that's centered about the initial point clicked.

As we move the mouse cursor (a cross showing we are in the inserting circle tool) near the point the graticule jumps to the point because snap to points mode is on.

We click and hold the click. The cursor jumps directly to the point. While holding the click we press the Insert key on our keyboard to launch dialog mode.
The **Ellipse** dialog pops up. Since we are already happy with the center position clicked for the new circle, all we need do is specify the **Width** and **Height** of the circle, using **3000** in both boxes. Since the drawing is in **Orthographic** projection that uses meters we are specifying a circle (since both width and height are the same) with a radius (one half of the width) of **1500** meters. We press **Apply**.

Instantly, a circle is created of exactly the width and height (and thus, the radius) we wanted.

This example shows a useful combination of several Manifold capabilities. We began by choosing a visual tool that makes obvious sense how to use (click a point and drag to create a circle). We used **snap** mode to click exactly on a central point of interest and then we shifted gears, pressing the **Insert** key to launch dialog mode. In dialog mode we specified exactly the size of circle desired.

There is actually one more technique in use that an experience user would apply when setting up this task, and that is the decision to project the drawings in use to **Orthographic** projection so that the units of measure available for specifying the circle's size would be meters.

If we had wanted to specify the size of the circle in feet we could have used a projection that used feet as the linear measure in the projection's coordinate system. Or, if we had wanted to specify a circle or ellipse using left, right, upper and lower bounds that were latitude and longitude degrees we could have used latitude / longitude projection for which the coordinate system units are geographic degrees.

**Rectangles**
Pressing **Insert** when using a visual rectangle tool when editing, painting or selecting will pop up the **Rectangle** dialog. The dialog allows entering the coordinates of the center of the rectangle and the dimensions.

We operate the dialog by entering the parameters we would like and pressing **Apply** or by pressing **Continue** to return to visual mouse mode.

We can specify the rectangle of action either by providing the left, right, upper and lower bounds and pressing **Apply** or by providing the location of the center and the width and the height and pressing **Apply**.

---

### Rectangle

**Bounds**

- **L**: The X coordinate of the left edge of the rectangle.
- **R**: The X coordinate of the right edge of the rectangle.
- **T**: The Y coordinate of the top edge of the rectangle.
- **B**: The Y coordinate of the bottom edge of the rectangle.

**Center / size**

- **X**: The X coordinate of the center of the rectangle.
- **Y**: The Y coordinate of the center of the rectangle.
- **W**: The width of the rectangle in units employed by the coordinate system of the component.
- **H**: The height of the rectangle in units employed by the coordinate system of the component.

**Apply**

Apply either the **Bounds** settings or the **Center / size** settings to specify the rectangle in dialog mode.

**Continue**

Abandon this dialog and return to visual mode.
Parameter boxes in the dialog will be loaded with whatever coordinates are implied by the position of the mouse at the time the Insert key was pressed to launch dialog mode. For example, an "insert on center" form of an insert rectangle command for drawings will have the location of the central point clicked already loaded into the X and Y boxes.

**Lines and Areas**

Pressing Insert when using a visual tool when editing, painting or selecting will pop up the Coordinate dialog. This dialog also works in dialog mode with the Tracker distance measurement tool and the grabber panning tool. All of these tools in visual mode depend upon a series of mouse clicks to specify a series of coordinates that make up the line, define the area or specify locations clicked for the tracker or grabber.

The dialog allows entering the next coordinate desired in a variety of ways. It can be used to specify the next coordinate through direct entry of desired X,Y coordinates, as a difference in X,Y from the current position, as an angle and distance from the current location, or as a difference in angle and distance from the angle and distance of the last segment created.

We operate the dialog by entering the parameters we would like and pressing Add to add the coordinate and continue, by pressing Continue to return to visual mouse mode, by pressing Undo to undo the last coordinate added or by pressing Finish to add the coordinate and complete the overall tool action.

**Coordinate**

<table>
<thead>
<tr>
<th>X / Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>X: 60631.59594</td>
</tr>
<tr>
<td>Y: 49903.80004</td>
</tr>
</tbody>
</table>

**Delta X / Y**

| X: -26.2952604 |
| Y: 26.2952604 |

**Angle / distance**

| A: -45.00 |
| D: 37.1671140 |

**Delta angle / distance**

| A: -45.00 |
| D: 37.1671140 |

| Add |
| Continue |
| Undo |
| Finish |

X / Y: Use this section to specify the coordinate by providing its X and Y coordinates.

X: The X coordinate of the coordinate.
Y The Y coordinate of the coordinate.

**Delta X / Y** Use this section to specify the coordinate by providing the change in X and Y directions from the previously specified coordinate.

X The change in X direction from the previous coordinate.

Y The change in Y direction from the previous coordinate.

**Angle / Distance** Use this section to specify the coordinate by providing the angle and distance to the new coordinate from the previous coordinate.

A Angle - Angle in degrees to the next coordinate.

D Distance - Distance in units employed by the coordinate system of the component to the next coordinate.

**Delta Angle / Distance** Use this section to specify the coordinate by providing the difference in angle and distance to the new coordinate from the angle and distance used to define the previous coordinate.

A Angle - Difference in angle in degrees to the next coordinate.

D Distance - Difference in distance in units employed by the coordinate system of the component to the next coordinate.

Add Add this next coordinate using the settings in the X / Y, Delta X / Y, Angle / distance or Delta angle / distance boxes.

Continue Abandon this dialog and return to visual mode.

Undo Undo the creation of the previous coordinate and reset the dialog to the values that prevailed just before its creation.

Finish Complete the tool without adding the current coordinate. Used, for example, to complete the creation of a new line when the last coordinate has been added.

Parameter boxes in the dialog will be loaded with whatever coordinates are implied by the position of the mouse at the time the **Insert** key was pressed to launch dialog mode. Since the visual mouse tool must be clicked at least once to start the tool the starting position, that is, the first coordinate, is already known so that the “delta” boxes and angle / distance boxes can be used.

**Example**

In this example we will create a line using dialog mode. We will be using a drawing in Orthographic projection so that units of measure will be meters.

We begin by choosing the **Insert Line** tool. This tool is used interactively to draw lines by using the mouse to click a series of clicks for each coordinate of the line followed by a right click to end and create the line.

We click into the drawing at the starting point of the line and then press the **Insert** key on the keyboard.
This pops up the **Coordinate** dialog, which is pre-loaded with the X and Y starting position at which we clicked. The drawing is projected so the starting X, Y coordinates are in linear measures and not in latitude / longitude degrees.
We will define the next coordinate for the line using the **Angle / distance** parameters to specify a bearing of **45** degrees and a distance of **250**. Because this drawing is in **Orthographic** projection the distance of **250** means 250 meters. :Press **Add** to add the coordinate implied by that bearing and distance.

![Coordinate input](image)

This extends the proposed line to the coordinate that is 250 meters away from the starting point at an absolute bearing (angle) of 45 degrees.

We will now continue the line using a slightly different technique.
For the next coordinate we will use a relative angle and distance using the Delta angle / dist boxes. We provide a relative angle of 45 and a distance of 0. We are telling the system to draw the next coordinate at a relative angle to the previous segment of plus 45 degrees at a distance that has zero change over the previous distance used.

The result is that the next segment is the same distance as the previous segment, 250 meters, and is drawn at a 45 degree angle to the previous segment. We could have achieved the same effect using the Angle / distance boxes with an angle of 90 (the absolute bearing of the desired line) and a distance of 250.

The difference between Angle / distance and Delta angle / dist is like the difference between walking a path following absolute compass directions and following relative directions.

- **Angle / distance** is like following instructions phrased like "On a compass bearing of 45 degrees walk 250 meters. Next, on a compass bearing of 90 degrees walk 250 meters. Next, on a bearing of 135 degrees walk 250 meters.
- **Delta angle / dist** is like following directions phrased like "Walk 250 meters in the direction you are headed. Turn clockwise 45 degrees and walk the same distance. Turn clockwise again another 45 degrees and walk the same distance again."

In the first case above which way you head for each segment does not depend upon what happened before. In the second case the direction you head depends upon the direction you were heading the previous segment.
We will continue extending the line by pressing the **Apply** button for **Delta angle / dist** to place the next coordinate using relative angle and direction.

The next coordinate for the proposed line is thus placed at a distance of 250 meters at an angle of 45 degrees to the previous segment.

We can continue pressing the **Apply** button for the same **Delta angle / dist** setting of 45 degrees and zero change in distance to create a series of segments 250 meters long that are at 45 degree angles to each previous segment. It may come as a surprise that doing so creates an octagonal shape like a stop sign.

To finish drawing a line we can click **Continue** to drop back into visual mouse mode and continue editing from there, or we can click **Finish**.
Clicking **Finish** ends the session with the equivalent of a right click on the mouse.

The result is that we have created a line that begins and ends at the same spot. Note that this is not an area, but we could have just as easily created an area using the same technique.

**Control Points**

Pressing **Insert** when inserting a control point will pop up the **Coordinate** dialog for control points.

<table>
<thead>
<tr>
<th>Coordinate</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>X / Y</strong></td>
<td></td>
</tr>
<tr>
<td><strong>X</strong></td>
<td>103.3102E</td>
</tr>
<tr>
<td><strong>Y</strong></td>
<td>19.908262Z</td>
</tr>
<tr>
<td><strong>Apply</strong></td>
<td>Add this next control point using the settings in the <strong>X</strong> and <strong>Y</strong> boxes.</td>
</tr>
<tr>
<td><strong>Continue</strong></td>
<td>Abandon this dialog and return to visual mode.</td>
</tr>
</tbody>
</table>

**Status Bar Reports**

When moving the mouse in creating a vector shape using visual tools such as the insert line tool, the status bar reports the length and angular bearing of the current segment as well as the location of the mouse cursor. Pressing **CTRL** will convert the report to use of alternate units (English or Metric) and will also enable use of ellipsoidal trigonometric calculations for the report.
For example, when creating a line as seen above...

\[-122.4364 \ 37.5500 \ L: 75.1393 \ m \ A: 43.88\]

...the status bar will report the mouse cursor location followed by L: length and A: angular bearing. The mouse cursor is located 75.1393 meters from the beginning of the segment at an angle of 43.88 degrees. The mouse cursor is also located at the latitude and longitude indicated.

**Tech Tip**

Using the backspace key when in dialog mode is equivalent to pressing an **Undo** button and undoes the last coordinate.

**See Also**

- Selection
- Adding Shapes
- Editing Objects
Drawings and Databases

One of the most important things about drawings is that the objects they contain, that is, every point, line or area in the drawing, can be tied to database information. This is done in a very simple way: every drawing has a table associated with it. Every object in the drawing is tied to one record, that is, one row, in the table. Even if there is no data associated with a given object, it still has a row in the table, an empty row if there is no data.

When a drawing is imported or created, a table will be created for it automatically. This table will appear under the drawing in the project pane in a hierarchical view.

Because every object in a drawing is associated with a record in a database table we can use visual methods in drawings to grab data from the table, and we can use tabular / database methods within tables to grab objects in the drawing.

Clicking on objects in a drawing is often a much easier way to find data than sifting through a text table. For example, it is much easier to select a group of Western states by clicking on them in a map than to search through a text table to find entries for "California", "Arizona", "Utah" and "Nevada."

Conversely, there are times when it is much easier to select items by clicking on their records in a table. We might sort a table by a population field and then select the top three or four states by population. We could then see them highlighted in red selection color on our map.

At times we may combine both methods, perhaps making a selection using table-oriented database methods and then refining the selection through mouse-based selection moves in the drawing.

Example

Imagine we have a drawing called Western_States that shows Western states in the US drawn as areas:

![Western States Map]

In the project pane this drawing appears with its associated table indented underneath it.

If we double click open the table for this drawing we might see that it contains four records, one for each state.
The table contains a field created and maintained by Manifold, the object ID number, plus three other fields. These other fields give the name, FIPS (Federal Information Processing Standard) code and the two letter postal abbreviation for each state.

When a drawing is created or imported Manifold automatically assigns a unique object ID number to each object. Manifold uses this object ID number to maintain the connection between each object in the map (the state areas in this case) and the corresponding row in the database table. Simple drawings might not have any data in their tables other than the object ID for each object. Drawings that are rich in data may have many fields in their database tables.

The association between objects in drawings and their record (row) in the database table is invisibly and automatically maintained for us by Manifold at all times. Whenever we select an object in the drawing we automatically select the corresponding row in the table. Whenever we select a row in the table we automatically select the corresponding object in the drawing. If we right click on the object and choose Fields in the context menu, we can see the object's fields.

If we imagine each area object as a shape cut out of cardboard, it is as though we have written the database information on the underside of each area. It is as if we could pick up the area for Utah and find written underneath the database information for that area. The association is that closely maintained.

In the example above, where do the fields come from? Our example uses a drawing originally published with many fields by the US Bureau of the Census. The drawing was imported, the drawing's table was opened and then some fields were deleted from the table to create the example shown. Most drawings used in GIS originate in some GIS data file that already has data fields created and filled in with values for the various objects in the drawing. In other cases we will use Manifold tools to merge data from databases into a drawing's table. In some cases, we will create a new field and add data by hand to each object.

Newcomers to GIS often focus on the visual presentation of maps. In contrast, experienced GIS users often feel that the most important aspect of GIS is the ability to use visual geography as a "handle" to grab and manipulate data within databases. The emphasis of much GIS work is not so much the precise drawing of cartographic features as it is amassing and utilizing rich databases of information within tables that are connected to maps. At times it is almost irrelevant how accurate or detailed is the cartography. For this reason, database skills and rich databases are extremely important in much GIS work.

Manifold includes a very fast and powerful database engine, so one can work with drawings that include many thousands of objects with many database fields per object in the drawing's table. As discussed in the Tables and Queries topics, Manifold's ability to manage database tables exceeds that of many so-called database management systems.
Manifold also has the ability to establish and exploit very rich connections with the world's premium DBMS products. Although an external DBMS product (such as Access, SQL Server or Oracle) is not required for using Manifold System, Manifold is capable of highly sophisticated usage of such DBMS products.

Except for linked drawings or relations in tables, tables that belong to drawings always exist entirely within Manifold. The data they contain is kept within the Manifold project file. Tables that are imported into a Manifold project are also managed within Manifold and their data saved within the Manifold project file. When tables are imported into the project, a copy of their data is taken from the original source and written into the project file.

Manifold can also create linked tables or linked drawings (which have their own tables that are also linked tables) in a project from external sources. When tables are linked into a project the data remains outside of Manifold in the original source. If a relation in a native table brings data into that table from a linked table, then that data also will come from the external source of data for that linked table.

Manifold has the ability to link to external tables for two main reasons. The first reason is simple space saving. At times it is more convenient not to make a copy of a table by importing it into the project. The second reason is dynamic updating by external programs. Some types of data, such as the count of customers in a particular county at exactly a certain moment, might be updated on a real time basis within a multi-user software application that keeps its data in a non-Manifold system such as SQL Server. If we link such a table into a Manifold project we can have the benefit of a table that appears in the project but which is dynamically updated by processes outside of Manifold.

**Using Relations between Tables**

A drawing's table can contain a large amount of data for objects in the drawing. However, at times it is more convenient to organize data using multiple tables instead of placing all fields in a single table. We may do this entirely within a Manifold project just to better organize our data. We could also use multiple tables because some of those tables need to be linked into the Manifold project from external sources in order to save space or to provide dynamically updated information.

A drawing's table can be used in database work just like any other table. In particular, it may be connected to other tables using relations. A relation between tables uses a key field with values common to both tables to connect the two tables so that columns from one table may appear in another table. This allows data that appears in one table to also appear in another table without having to physically duplicate the data.

To connect another table to a drawing's table, Manifold needs to know how the two tables should be related. This is done using a "key field" that contains matching values in the drawing's table and in the other table. This means that a drawing's table must have at least one field that can be used to match values with other tables.

For example, suppose our drawing of four states has only one field in the table in addition to the internal object ID field. The Abbrev column gives the two-letter US postal code abbreviation for each state:

<table>
<thead>
<tr>
<th>StatesTable</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
</tr>
<tr>
<td>32761</td>
</tr>
<tr>
<td>32762</td>
</tr>
<tr>
<td>32763</td>
</tr>
<tr>
<td>32775</td>
</tr>
</tbody>
</table>

Suppose also we have a state_info table, perhaps imported from some demographic database we found on the web, that gives information for each state in the US. Suppose this table also contains the two letter postal code for each state:
In this case, we can use the state postal code abbreviation as a key field to form a relation between the two tables. We can then “borrow” fields from the state_info table and include them in our drawing’s table as if they were fields native to that table:

<table>
<thead>
<tr>
<th>Name</th>
<th>State...</th>
<th>Abbrev</th>
<th>FIPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minnesota</td>
<td></td>
<td>MN</td>
<td>27000</td>
</tr>
<tr>
<td>Washington</td>
<td></td>
<td>WA</td>
<td>53000</td>
</tr>
<tr>
<td>Idaho</td>
<td></td>
<td>ID</td>
<td>16000</td>
</tr>
<tr>
<td>Montana</td>
<td></td>
<td>MT</td>
<td>30000</td>
</tr>
<tr>
<td>North Dakota</td>
<td></td>
<td>ND</td>
<td>38000</td>
</tr>
<tr>
<td>Michigan</td>
<td></td>
<td>MI</td>
<td>26000</td>
</tr>
</tbody>
</table>

The illustration above shows the drawing’s table after it has had a relation formed and three additional columns included from the state_info table. We can use the data borrowed from other tables just as if the included fields were built into the drawing’s table. Whenever we select the Utah area in the drawing, it is as though that area is a handle not just to the original state abbreviation but to all of the fields.

For example, we could tell Manifold to create labels that showed the contents of various data fields for each object. These labels could grab data from the included fields as well as from the original field in the table:
The data required will automatically come out of whatever tables are connected to the drawing's table via relations. Note that to make this magic work all we need is a key field that exists in the drawing's table that can be used to match up records with a corresponding key field in the other table.

Note also that since Manifold projects can contain linked tables that use data residing in external database providers, a drawing's table can have a relation with a linked table. If the external data and hence the linked table changes, the data shown in the map will also change. This is a good way of using table data in a drawing when the data changes frequently.

For example, if we wish to display weekly sales totals in a map of sales regions we could link the sales table in our corporate SQL Server database into our project and then form a relation between that table and a drawing's table that shows sales regions. Whenever we display the map using that drawing, it will automatically include the latest sales figures as they are kept in the SQL Server database.

We can use linked tables with simpler technology than SQL Server. For example, a linked table could be created from an Excel spreadsheet that is regularly updated to incorporate the latest sales figures.

**Tables Work with Lines and Points, Too**

The above examples use areas because they are easy to illustrate as having information written on them. Any map object, including lines and points, can have database information attached. Points are often used to show the locations of facilities, customers, towns, stores and other locations. Each point could have many data fields in the table.

Lines can have data fields in tables as well. Typical uses of fields attached to lines are to maintain information on roads or to record data for streams or other waterways.

**Tech Tip: Keeping Organized**

Each object in the drawing corresponds to a record (one row) in the drawing's table. Every object in that drawing will have the database fields in that table. Not every cell in every record has to be full of data, of course, but all objects in the drawing (that is, every row in the table) will have the field structure for that table.

To keep things organized it is wise not to mix up objects in one drawing that really would be better organized in separate drawings. For example, suppose we have a drawing of states as areas with demographic information for each state. If we also have a drawing of roads with maintenance data for each road, it would make sense to keep the roads and the states in separate drawings so that the two different drawings can have different fields in their tables. We can always combine the multiple drawings in a single map if we want to see the states and the roads together in one map.

Note that there's no problem mixing lines and areas in the same drawing; however, it's usually the case that objects shown as areas (like states) tend to be associated with different types of database fields than objects that are drawn with lines (like roads). An exception to this trend is the display of water features in "hydrography" layers in maps where water features are frequently drawn using both areas (lakes) and lines (streams) in the same drawing. Drawings showing hydrography features will often use the same database table for both lakes and streams, since the same information (name, fresh or salt water, perennial or seasonal feature, etc.) fields are often used for both.
Summary

- Every drawing has a table that at least contains one field, the object ID field.
- Every object in a drawing is associated with one record in the table. A record is one row in the table.
- Each data attribute field is one column in the table.
- All objects in the drawing will have all the fields in the database table associated with that drawing.
- When objects logically should have the same data fields, group them together in a drawing.
- Use maps to show different drawings together at the same time.

For More Advanced Users

This topic discusses tables associated with drawings in an elementary way, mostly using the word “database” to mean the internal storage of tables within a Manifold project using Manifold’s own internal database capabilities. There are two more advanced notions regarding drawings and databases we will develop as we get further into Manifold.

The first notion is that the tables that appear in Manifold projects, whether they are used by themselves or joined to a drawing’s table via a relation, can be linked into the project from some DBMS server or database file outside of Manifold. The next topic, Drawings and Tables, as well as topics in the Tables chapter develop this idea.

The second notion is that Manifold can store drawings themselves (as well as tables, images and surfaces) within some external DBMS server outside of Manifold. For example, the Western States drawing used as an example could be stored not inside a Manifold project but instead might be exported into a DBMS server such as SQL Server, Oracle, DB2, PostgreSQL, MySQL or some other DBMS. Manifold users could then link that drawing into their projects so that more than one person at the same time could view and edit that drawing. This idea is more fully developed beginning with the Spatial DBMS topic in this Introduction chapter.

See Also

The Query toolbar is a fast way to select objects in drawings using the data values they contain.

Drawings and Tables

Attaching Linked Tables to Drawings
Drawings and Tables

This topic gives specific examples of interacting with drawing tables, to show how the association between drawings and database tables discussed in the Drawings and Databases topic works in practice.

Recall that every drawing in Manifold has a database table tied to it. Each object in the drawing corresponds to a record (one row) in the database table. Every field in the database will be a column in the database table. A drawing’s table will have at least one field, the object ID field, which provides an identification number for each object in the drawing. The object ID is a unique number generated by Manifold when the drawing is created or imported. Tables can contain other fields as well. For example, a drawing showing cities as points could have a Name and Population field in the table that gives the name of each city and its population.

A drawing’s table can be used like any database table. For example, it can be used to form relationships with other tables, including tables that are saved outside of Manifold in external database providers.

When drawings are imported from a GIS format that saves both object and database table information the new drawing that appears in the project will automatically have a table created that is populated with data.

If we import a drawing from a format, such as AutoCAD DXF, that does not save database table information, or if we create a new, blank drawing in a project, Manifold will automatically create both the drawing and a table to go with that drawing.

Suppose we create a new drawing in the project and name it Drawing1.

By default, the minimum field in drawing’s table is an ID field that gives the object ID number of each object in the drawing. ID numbers are assigned on a unique basis for each new object created or imported into any drawing. All ID numbers are guaranteed unique.

We can open the drawing in a drawing window and also open the drawing’s table in a table window. When a new drawing is created there are no objects in the drawing or associated records in the table. To fit conveniently as screen shots within this documentation, both the drawing window and table window will be shown as relatively small windows.

Note that the ID field cells in the table are shown using the default light gray color that indicates read-only cells. The ID field in a drawing’s table may not be edited since the ID is the hard-wired connection between each object in a drawing and the record in the drawing’s table that corresponds to that object.

Drawings and their Tables are Synchronized
For drawings and their tables, any changes in one component are immediately updated in the other component. Adding or deleting an object in the drawing will add or delete the associated record in the table.

If we use the Tools toolbar to add an area to the drawing a record appears for it automatically in the table as well.

Note that the ID field cells in the table are shown using the default light gray color that indicates read-only cells. The ID field in a drawing's table may not be edited since the ID is the hard-wired connection between each object in a drawing and the record in the drawing's table that corresponds to that object.

Tables usually show more interesting data than Object IDs; however, as the simplest possible example of a drawing's table we will show just the Object ID field.

Creating another area in the drawing adds another record in the table as well.
Creating a third area adds a third record, shown with an ID of 3. [Note: It is very rare that we would ever see such low numbers in sequence as object IDs since we normally work with large, complex maps where objects are imported, deleted, created and so on in potentially large numbers. Object IDs are therefore more normally semi-random numbers and not neat, small numbers in sequence.]

The screen shot above shows our drawing and table after we have used the Tools toolbar to add several points. See the Adding Points, Lines and Areas topic for information on how to add points to a drawing.

Using the Tools toolbar we can add two lines as well. Note that two more records have appeared in the table.
Selections are Synchronized

A drawing and the drawing's table share a single selection since both the drawing and the table are just two different ways of seeing the same objects. Any changes to that selection in either the drawing or the drawing's table will immediately be updated in the other.

Any Selection made in the drawing will simultaneously appear in the table with a change in background color of the associated records. Selected records will have a faint red background color by default. The example above shows we have selected two areas, a line and a point in the lower half of the drawing.

If we change the selection by selecting two points and an area in the upper part of the drawing the selection shown in the table will immediately be updated to match.
The Selections pane shows the same set of saved selections for both drawings and their tables. The example above shows the Selections pane opened and moved into position so it can participate in our screen shots. We have selected the three areas and have saved them as a saved selection called Areas.

We can select the two lines and save them as a saved selection called Lines. [Note: there is nothing magical about our choice of the word Lines for the name of this saved selection other than the mnemonic value to remind us at a glance of what comprises that selection. We don't have to use the word Lines to name a saved selection that consists of lines. We could just as easily call it Fred or Bob or Points if we wanted to choose a confusing name.]
At any time, we can click into the Selections pane and get a Preview of any of the saved selections by pushing in the Preview button in the Selections pane. When the Preview button is pushed in the highlighted saved selection will be shown in blue preview color in both the drawing and in the table. If a record is both in the current selection and in the preview it will be shown in a color mid-way between the red selection color and the blue preview color. [Manifold will automatically interpolate between whatever selection color and preview color we specify in the Tools - Options dialog].

We can use the selection command buttons to combine a saved selection with the existing selection. In the illustration above we have pushed out the Preview button, clicked on the Areas saved selection and then pressed the Select Add button in the Selections pane toolbar. This adds the areas to the current selection for a selection that consists of both lines and areas.
Deletions are synchronized as well. Suppose we select just the areas and choose **Edit - Delete**.

The areas will be deleted in the drawing and their associated records will be deleted in the table as well.

**A Drawing’s Table Cannot Add Objects to the Drawing**

We cannot add objects to a drawing by adding records to the drawing’s table. The reason is that the information seen in the table is only part of the information that defines the object. An object in a drawing is defined by the geometric coordinates that specify its location and shape. These are not visible or part of the table: they are part of the data within the drawing itself.

Deleting an object in a drawing will delete the object’s record in the drawing’s table. Deleting a record in a drawing’s table will delete the associated object in the drawing.

**Note:** It is possible to create a linked drawing that is created based on the contents of a table, either using geocoded tables or using geometry fields in tables. In that case, the table data controls both the linked drawing and the linked drawing’s own table. If a record is added to the source table then a new point will be created in the linked drawing and a new record will appear in the linked drawing’s own table. However, even in that case we can’t add objects to the linked drawing by adding records to the linked drawing’s own table.

**Attaching External Tables to a Drawing’s Table**

So far we’ve seen that every drawing is linked to a table and that the drawing’s table is created within the project at the same time the drawing is created. All such tables exist entirely within the project. Sometimes we would like to link our drawing’s table to an external table that is linked into the project.
To do this we use relations with linked tables. See the Attaching External Tables to Drawings topic.

**The Info Pane**

A convenient way of seeing data from drawing tables one record/object at a time is to use the Info pane to step through data one object at a time.

**Tech Tip**

The illustrations above show the first object created having an **ID** of **1**. Although the first object created in a new drawing may well have an **ID** of **1**, this is not guaranteed. Object IDs are unique within a given .map file but are not guaranteed to begin with **1**. It is therefore entirely possible to create a new drawing and to then create a new object in that drawing that has an object ID of, say, **5**. Even in the case of the first drawing in a completely blank drawing it is still possible not to get an object ID of **1** for the first object created.

**See Also**

The Query toolbar is a fast way to select objects in drawings using the data values they contain.
Working with Tables

This introductory topic provides a brief overview of how Manifold approaches tables and databases. It is intended for readers already familiar with database concepts such as tables and queries. See the Tables topic and subsequent topics for a complete guide to working with tables. See the Working with Queries topic for an introduction to queries. See the Query toolbar topic for a fast way to select records in tables.

Tables show data organized into rows and columns. Every row in a table is a record. Every column in a table is a field. We will use the words "row" and "record" interchangeably. We will also use the words "column" and "field" interchangeably.

### Tables and Drawings

Every drawing in Manifold has a table associated with it. Each object in the drawing (that is, every point, line or area in the drawing) is linked to a row in the table. Drawings have a table even if there are no data fields saved for each object. In that case, the drawing's table is mostly empty. It has only one field, the object ID field, for each row. The object ID field is used to link each row to its associated object and cannot be edited.

Deleting an object in a drawing will delete the object's record in the drawing's table. Deleting a record in a drawing's table will delete the associated object in the drawing.

### Other Tables

Tables may also be database tables having nothing to do with drawings that are imported into a Manifold project or linked into the project from an external database source. New tables may also be created with Manifold.

Tables may be native tables (also called local tables) that have at least one column that exists entirely within the project, or they may be linked tables that are created from other tables. Linked tables are created from other tables, which could be within the same project or which could come from data sources outside of Manifold. Native tables can also use data from other tables if they include relations with other tables.

Tables may be imported or linked into projects from almost any database, even from those that have nothing to do with drawings or maps. We can import or link tables from a vast range of different types of data files and data sources, from ordinary text files such as CSV files, from applications files such as Excel XLS files that contain data, and from a very wide range of database connections and database providers such as Access, SQL Server, Oracle, IBM DB2, MySQL and the like. It's actually very difficult to find a source of data that a knowledgeable Manifold user cannot utilize.

We will often work with tables that have no geographic context. We may wish to prepare data for later use together with drawings and maps, or we might simply wish to use Manifold as a general-purpose means of viewing, analyzing, exploring and managing databases.
The illustration above, for example, shows a project with tables imported from the Microsoft Northwind Traders example database shipped with Microsoft Access.

If we open the Customers table we see the same data familiar to many Access users who have worked with this sample database.

There are two types of sources for database tables that may be imported or linked into a Manifold project:

- Tables from files and file types that may be opened directly by Manifold's own database engines and so require no other database software. These include .mdb, .dbf, .udl, .wkX, .xls, .db, and .html files as well as ASCII or text database files organized as comma separated value (.csv) form. Such file-based data storage options appear directly within the Files of type box in dialogs when importing or linking.
- Tables from data sources that are provided by external database engines via OLE DB, ODBC, ADO .NET, Oracle Call Interface (in the case of Manifold Enterprise Edition) or similar connections to database providers. Some such sources may require external database software. The default Manifold installation process will include a default set of standard Windows drivers for popular databases such as SQL Server and Oracle. Such data sources are accessed using the Data Source dialog, which appears when the Data Sources () choice is used within the Files of type box in dialogs when importing or linking.

To import or link a table into a Manifold project

1. Choose File - Import - Table or File - Link - Table depending on whether you wish to import the table or to link it.
2. In the Files of type box choose the access method desired. Choose one of the standard file types, such as MDB or XLS to connect directly to common file types. Choose the Data Sources () item to connect via ODBC, OLE DB, ADO .NET or a native connection to a DBMS server.
3. Browse over to the database file and open it. See the Data Source dialog topic for examples connecting using that dialog.
4. Choose the tables to be imported. Choose more than one table if desired.
5. When linking tables, check Read Only if you wish a read-only link. Some connection technologies such as ADO .NET or some ODBC providers may allow only a read-only link.
6. Press OK.

After the initial choice of File - Import or File - Link, all dialogs are the same. The only difference is whether the data in the table is copied into the project or if it is left outside in an external file or data source.

Alternate method to import or link a table
1. Open the Tools - Database Console dialog.

2. Next to the Data source box, click on the [...] button to launch the Data Source dialog. Connect using that dialog as shown in the the Data Source dialog topic.

3. The upper pane will show the contents of that data source, including any tables and the fields they contain. Click on a table to highlight it.

4. Click on the Import button in the dialog's toolbar to import the highlighted table. Click on the Link button to link the table.

5. Press Close.

In all cases, tables that are imported or linked will appear to be part of the project like any other table. Manifold can import data from or link to a very wide range of database files and database providers.

See the Data Source dialog topic for information on using that dialog to connect to databases.

See Importing and Linking Tables for more information on using existing tables.

See Creating New Tables for information on creating new tables.

See the Database Console topic for a useful console for browsing databases and importing and linking tables.

The database console can also be used to execute SQL commands within an external database provider, such as SQL Server or Oracle.

**Table Windows**

Clicking open a table will display it in a table window, a classic row and column text database table presentation. A table may be opened in more than one window simultaneously. Opening an SQL query will also result in a table showing the results of the SQL query.

<table>
<thead>
<tr>
<th>ID</th>
<th>Place name</th>
<th>SOM</th>
<th>VIVIEN...</th>
<th>BUS...</th>
</tr>
</thead>
<tbody>
<tr>
<td>27</td>
<td>AQUASCALIENTES</td>
<td>2374.72</td>
<td>130599</td>
<td>1031</td>
</tr>
<tr>
<td>8</td>
<td>BAJA CALIFORNIA</td>
<td>29614.39</td>
<td>373476</td>
<td>2549</td>
</tr>
<tr>
<td>9</td>
<td>BAJA CALIFORNIA SUR</td>
<td>20732.4</td>
<td>68479</td>
<td>375</td>
</tr>
<tr>
<td>21</td>
<td>CAMPECHE</td>
<td>22205.42</td>
<td>110259</td>
<td>635</td>
</tr>
<tr>
<td>31</td>
<td>CHIHUAHUA</td>
<td>23054.57</td>
<td>596696</td>
<td>2383</td>
</tr>
<tr>
<td>12</td>
<td>CHIHUAHUA</td>
<td>98147.57</td>
<td>549541</td>
<td>2430</td>
</tr>
<tr>
<td>14</td>
<td>CHIHUAHUA</td>
<td>20414.52</td>
<td>441252</td>
<td>1272</td>
</tr>
</tbody>
</table>

Table windows show many records at once. Selected records appear highlighted in red selection color. Records will be shown in selection color whether they were selected within the table window or (in the case of drawing tables) if they were selected by clicking on the corresponding objects in the drawing. Records may also be selected using the query toolbar or queries.

Table windows are controlled using a mix of menu command choices, context menus, selection and use of the transform toolbar. Try right clicking onto a column head, a record handle, and a cell to see the different context menus that appear with tables. See the Tables - Context Menus topic for a summary guide to context menus in tables.

By default, table windows do not use horizontal or vertical gridlines between rows and columns. Gridlines may be turned on in Tools - Options.

**Columns**

Columns are headed with the name of the field they show. We can reorder, show or hide columns in a table using the View - Columns command. Using the View - Columns command, we can also display intrinsic fields as columns in the table. Intrinsic fields are computed quantities, such as the length of lines, or system quantities such as the latitude and longitude position of a point. Right click onto a column head to see a context menu of many additional commands that work with columns.

The boundary lines between column heads may be moved left or right to make a column wider or narrower. If you are using Windows 2000 or Windows XP columns may be dragged and dropped into different left to right arrangements. We can also use Best Fit or Best Fit All to automatically set the width of columns.
Click on a column head to sort it in ascending (A to Z) order. SHIFT-click on a column head to sort it in descending (Z to A) order.

**Specifying Languages for Columns**

Columns in tables can have different languages specified. This allows one column to show text in one language while another shows text using a different language character set. Several different languages can be used in different columns within the same table. By default, tables use the Windows locale setting for the current user. If desired, any text column in a table can be set to a different language by right clicking on the column head and choosing **Language**.

**Rows**

![Row selection](image)

**Records** have a **record handle** at their left margin. A black triangular arrow marks the current record in its record handle.

<table>
<thead>
<tr>
<th>U 11</th>
<th>Utilities Kasedaen</th>
<th>Herrmee Maeninen</th>
<th>Owner</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARIS</td>
<td>Paris specialties</td>
<td>Marie Bertrand</td>
<td>Owner</td>
</tr>
<tr>
<td>PFR</td>
<td>commercial clients</td>
<td>Guillermo Fernandez</td>
<td>Sales Rep</td>
</tr>
</tbody>
</table>

The **current record** will be specified by a current record arrow in its handle as well as by outlining the active cell.

<table>
<thead>
<tr>
<th>U 11</th>
<th>Utilities Kasedaen</th>
<th>Herrmee Maeninen</th>
<th>Owner</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARIS</td>
<td>Paris specialties</td>
<td>Marie Bertrand</td>
<td>Owner</td>
</tr>
<tr>
<td>PFR</td>
<td>commercial clients</td>
<td>Guillermo Fernandez</td>
<td>Sales Rep</td>
</tr>
</tbody>
</table>

When selected, the **current** cell is highlighted by a slightly lighter shade of the red selection color.

**Selection Methods**

Selecting records in the Table window is part of many tasks in Manifold. Selection in tables, like drawings, uses selection modes to specify whether the selection command should replace, add to, subtract from, invert with or intersect any previously existing record. The default mode is **Replace**.

- Click on a cell to make it the current cell and to make that record the current record.
- Click on a record handle to select that record.
- SHIFT-click on a record handle to select the record and to also select all the records between it and another selected record. This is often used to select a series of records: click on the top record and then SHIFT-click on the bottom record and all the records in between will also be selected.
- CTRL-click on a cell to select that record.
- SHIFT-CTRL-click on a cell to select a range of records.

See the **Selection in Tables** topic for more details.

**Copy and Paste Operations with Columns**

Copy and Paste are handy commands for copying data between fields. Suppose we have a numeric field that is a **double**, and we would like to convert the field into an **integer**. We do this by first creating a new field that is an integer. A new column will appear. Next, we **Copy** the contents of the double numeric field's column and then **Paste** into the new integer field's column.

This method works between reasonably compatible types. For example, numbers like **3.1415** may be converted into text strings such as "3.1415", but it does not make sense to attempt to convert "Main Street" into a number.
We can also use the Transform toolbar to copy values from one column to another, also with automatic conversion whenever possible.

**Editing Cells**

See the topic Editing Data in Tables for details on changing values inside cells and adding new records to tables.

See the Editing Intrinsic Fields in Tables topic for a discussion of how we can move objects by editing their locations as they appear in intrinsic fields in a table.

**Bookmarks**

Bookmarks are icons in the left margin of text components such as comments, queries and scripts. They allow fast navigation within lengthy text. Bookmarks are also available within tables, comments, scripts and queries. Within tables they are very useful for marking particular records when jumping back and forth between records in a table. When tables or queries are sorted the bookmarks correctly travel with the records into the new sort order.

To set a bookmark in a table, click anywhere in the row to be marked.

Press F8 or choose **Edit - Bookmarks - Toggle** to set the bookmark. To clear the bookmark, press F8 or choose **Edit - Bookmarks - Toggle** again.

Add other bookmarks as desired. We can now use Ctrl-F8 to jump to the bookmark below the current row or Shift-Ctrl-F8 to jump to the bookmark above the current row. See the Edit - Bookmarks topic more on bookmark.

**Effect of Tools - Options Settings**

When importing tables, keep in mind that the default setting in Tools - Options is to not import empty columns. Check the **Import empty columns in tables** option to import tables with columns that do not contain any values in the originating table.

Note also the setting of **Trim strings imported from external databases** (on by default). This will delete leading and trailing whitespace characters from imported table strings.

The **Trim strings** option (just like the **Trim** table transform operators) removes characters listed in the Tool Properties pane’s list of token separators. By default, these are the “white space” characters consisting of the space character, tab, newline and carriage return. Note that adding any other characters to the separator list in Tool Properties will subject them to removal as well if they occur as leading or trailing characters.
**ViewBots** are one-line analytic instruments used to dynamically compute a statistical or comparative measure over a subset of records and are extremely useful for analyzing tables. They are one of the most popular functions in Manifold for experienced users. See the ViewBots topic.

**Transform Toolbar**

The Transform toolbar is an extremely powerful tool for managing and editing tables. It is easy to use and should be mastered by every Manifold user doing serious work with tables. New users and experts alike will experience dramatic productivity gains by learning to use the transform toolbar.

**Active Columns**

Active Columns are a way of embedding short (or long) snippets of scripts within tables to automatically accomplish calculations. The idea is similar to that of spreadsheets, but appearing in tables in a much more powerful way.

**Decision Support System**

Manifold's Decision Support System provides remarkable data mining capabilities by applying fuzzy inferences in an easy to use way.

**Primary Keys a Must with DBMS**

When working with server-based OLE DB providers such as DB2, SQL Server and Oracle, users are strongly encouraged to maintain primary keys in all tables linked into the Manifold project. A side effect of how such servers interact through OLE DB is that if the table does not have a primary key, performance will be greatly reduced.

**ODBC and .map Files**

Manifold includes an ODBC driver. This driver allows other applications to work with data in tables and queries in a Manifold .map project file. See the Using the Manifold ODBC Driver topic.

**Notes**

Tables within Manifold have many powerful capabilities within the context menus that appear when right clicking on different parts of the table. For example, one can change field types using the Change Type choice that appears when right clicking onto a column head. In another example, one can right click on a record and choose More Like This to automatically see more records like that one sorted to the top of the table. Take the time to read through the Tables topic and subsequent topics in the Tables book to learn about these and many other powerful capabilities.

Many of the illustrations for tables are screen shots made using the Nwind.mdb sample database provided by Microsoft. This is a version of the Northwind.mdb sample database distributed with Access and thus familiar to many Microsoft Office users. Nwind.mdb is provided on the Manifold System CD in the Help examples folder.
Working with Queries

Queries are components that contain statements written in the database language SQL. SQL statements are most often used to select data for viewing, but queries can also be used to alter the structure of tables or to alter the data tables contain. Queries can select data from one or more tables, filtering, grouping and ordering it as desired or otherwise managing the contents of tables that exist within a project. The simplest form of query chooses columns and records from one table and displays them as a table.

Queries are easy to create. Use File - Create - Query to create a new query and then edit it by double clicking on it to open it (or by right clicking on it and choosing Open). Enter the SQL statement you wish to use and close the query.

Run the query by highlighting it and clicking the Run button. Whenever the query is run it will execute the SQL statement. If the SQL statement in the query returns a table the query will display it in a table window. The resultant table may be used like any other.

SQL is a really easy way of phrasing queries. Don't be intimidated by experts who make SQL out to be more complex than it need be. While SQL can certainly be used to write fearsomely complex queries that no one understands, most of the time it is used by average people to write very simple, obvious queries this one:

```
SELECT [Contact Name], [Phone] FROM [Customers] WHERE [Country] = "France";
```

The table names and fields used in a query refer to whatever tables we have in our project. If we've imported the Customers table from the Nwind.mdb sample database into our project and place the above SQL statement into a query, when we open the query we will see a simple table with two columns:

<table>
<thead>
<tr>
<th>Contact Name</th>
<th>Phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Daniel Torini</td>
<td>30.59.84.10</td>
</tr>
<tr>
<td>Frédérique Côteaux</td>
<td>88.60.15.31</td>
</tr>
<tr>
<td>Martine Rancé</td>
<td>20.16.10.15</td>
</tr>
<tr>
<td>Janine Labrune</td>
<td>40.67.86.88</td>
</tr>
<tr>
<td>Arnette Roulet</td>
<td>61.77.61.10</td>
</tr>
<tr>
<td>Laurence lebihan</td>
<td>91.24.45.40</td>
</tr>
<tr>
<td>Paul Harrich</td>
<td>64.71.15.11</td>
</tr>
</tbody>
</table>

Simple SQL statements displaying data from a single table are written as follows:

```
SELECT (fields) FROM (table name) WHERE (condition)
```

SQL is not case sensitive except inside text strings used within quotes in criteria. Many SQL writers like to use CAPITAL letters for SQL key words, like the SELECT statement, in order to keep their queries better organized, but that is not necessary. We can use upper or lower case as desired. We can use parentheses to group criteria within criteria to keep our statements legible, and we can use tabs and spaces, multiple lines and indents as desired to keep things clear:

```
SELECT [Contact Name], [Phone] FROM [Customers]
WHERE [Country] = "France"
OR [Country] = "Germany"
```

As can be seen from the above, words like And and Or may be used within criteria.

A search of Internet using any good search engine looking for keywords such as "SQL", "style" and so on will produce numerous pages advocating various styles of writing SQL queries. Choose whatever writing style you like. Extra parentheses don't hurt. Use them to keep queries legible and well organized and to guarantee the order of precedence in comparison operators is what you want it to be in complex queries.

SQL is very powerful and even fun. Don't deny yourself the benefit of this easily learned tool. The nice thing about SQL is that it is easy to learn to write simple queries and it is very easy to extend one's capabilities in an
incremental way. With a little practice, you too will be spinning out intense little SQL queries as fast as your fingers can fly over the keyboard.

**Dot Nomenclature**

"Dot" nomenclature allows us to refer to exactly a desired field within a given table even when other tables in our project might use the same field name in a different way. Dot nomenclature is simply the table name followed by a period `.` and then the field name:

```
[Customers].[Contact Name]
```

The above refers to the **Contact Name** field in the **Customers** table. If we are writing simple queries that involve only one table we need not use dot nomenclature. We could simply write **Contact Name** as in the examples above. However, if we are working with more than one table we can always use dot nomenclature to clear up any ambiguities about which field is intended. Sometimes it is necessary to use dot nomenclature to resolve ambiguities.

**Field Names in Brackets**

Manifold as well as many modern database management systems allow field names to contain the space character and other non-alphanumeric characters. Such field names should be enclosed in `[]` square brackets when written in queries. Simple field names that do not contain spaces or other non-alphanumeric characters need not be enclosed in brackets.

To preserve maximum capability to export data to a wide variety of database systems we suggest the use of field names that do not contain spaces or other unusual characters.

**Bookmarks**

Bookmarks are icons in the left margin of text components such as comments, queries and scripts. They allow fast navigation within lengthy text. Bookmarks are also available within tables and queries. Within tables they are very useful for marking particular records when jumping back and forth between records in a table. When tables or queries are sorted the bookmarks correctly travel with the records into the new sort order.

To set a bookmark in a table, click anywhere in the row to be marked.

Press F8 or choose **Edit - Bookmarks - Toggle** to set the bookmark. To clear the bookmark, press F8 or choose **Edit - Bookmarks - Toggle** again.

Add other bookmarks as desired. See the **Edit - Bookmarks** topic for notes on how to jump between bookmarks or to go to the previous or next bookmark.
ODBC and .map Files

Manifold includes an ODBC driver. This driver allows other applications to work with data in tables and queries in a Manifold .map project file. See the Using the Manifold ODBC Driver topic.

Notes

See the Queries topic for more. That topic and the topics following it in the Queries book introduce various other useful SQL methods, including examples of how to create a new query table that displays data from two other tables at once. The more advanced topics also discuss “action queries” that are used to do things with the data.

This documentation uses examples drawn from the Microsoft Northwind.mdb and Nwind.mdb sample databases. The two are very similar, except that Nwind.mdb uses spaces in the names of fields for some tables.

Just about everyone in computing thinks that “SQL” is an acronym for “Structured Query Language.” That’s not true, as the ANSI standard defining SQL officially names it “Database Language SQL.” Experts will also point out that SQL is not structured, it is used for more purposes than just asking about things (the conventional meaning of the English word “query”) and it is technically not a “language” in the Turing sense of the word.

According to ANSI, “SQL” should be pronounced by saying the letters as in “ess cue ell.” However, many people prefer to pronounce the term as the English word “sequel,” especially when referring to Microsoft’s SQL Server DBMS product since “sequel server” is more alliterative and easier to say than “ess cue ell server.” It is very common to use both pronunciations, even in the same sentence, as in “This ess-cue-ell works in Sequel Server but not in Oracle.”

See Also

The Query toolbar is a fast way to select objects in drawings or records in tables using the data values they contain. For very simple queries the query toolbar is much faster than creating and running an SQL query.

See the Database Console topic for a useful console for browsing databases and importing and linking tables. The database console can also be used to execute SQL commands within an external database provider, such as SQL Server or Oracle.

See Selecting Objects with Queries for information on selecting objects within drawings or drawing layers in maps using SQL queries.

See the Geocoding Extensions and Raster Extensions and Spatial Extensions and topics for a list of extensions to SQL in Manifold that allow creation of SQL queries that make spatial comparisons or employ the geocoding engine for work with street addresses.
Finding / Replacing Data

Search and replace operations in tables are easy with the Edit - Find, Edit - Find Next and Edit - Replace dialogs. These are very similar to the analogous dialogs in Microsoft Office applications. These commands are available for table components.

To search for a value:

1. Open the table.
2. Choose Edit - Find.
3. Enter the value to be found (text, a number, or a regular expression) in the Find what box.
4. Choose optional settings, such as matching case.
5. Press Find Next to find and select for editing the next instance of the value, or choose Find All to find and select all instances of the value.
6. Searches proceed from the current position in the table towards the end of the table. To begin a search at the beginning of the table, press Restart.

Using Find in a drawing will find and select the object that has that value in its table record. Objects will be selected as the primary selected object, ready for editing. Once the Find dialog has been used, we can use the Edit - Find Next command to find the next instance of the previously found value.

Options in Find and Replace Commands

- **Find what**  The value or regular expression to be found.
- **Replace with**  Value used to replace found values.
- **Match whole word only**  Match the value only when it occurs as a whole word. With this checked, "the" will match the word the but not occurrences of these letters within words such as weather.
- **Match case**  Find and Replace are case insensitive by default. This forces a match exactly on the case given in the Find What value. Example: unchecked, "the" will match The and tHe.
- **Use regular expressions**  Regular expressions are a concise and flexible notation for finding and replacing patterns of text.
- **Search only in**  By default, Find and Replace will search all fields. Check this to restrict the search to only the field given.
- **Find Next**  Find the next instance of the search value.
- **Replace**  Replace this instance of the search value.
- **Replace All**  Replace all instances of the search value.
- **Restart**  Move the current record pointer to the beginning of the table. Searches proceed from the current record pointer only.
- **Close**  Close the dialog

Example

To find the province of Morelos in the drawing of Mexican provinces provided as a sample, we import the mexico drawing and then open the drawing's table. Choose Edit - Find to launch the Find dialog and then enter Morelos in the Find What box. Press Find Next to select this province.
We can replace values using the Edit - Replace dialog, which is the same as the Find dialog with the addition of a Replace with edit box and Replace and Replace All command buttons.

**Using Regular Expressions**

Regular expressions are a concise and flexible notation for finding and replacing patterns of text.

See the Regular expressions topic for information on using regular expressions. These are a much more sophisticated and flexible method than simply using the * and ? characters.
Copy and Paste As

The Windows clipboard Copy and Paste As commands are used throughout Manifold to copy data, to transfer data between components and to transform data from one form into another. For example, copy and paste within the project pane are used to convert components into different forms. To convert a surface into an image we Copy the surface component in the project pane and then Paste As an image component into the project pane. Subsidiary dialogs will be called on the fly as necessary with appropriate conversion options to make the desired image component out of the data in the surface.

Any standard Windows clipboard commands work for Copy and Paste As. For example, when a component is highlighted in the project pane, we can copy using Edit - Copy, by right clicking on the component and choosing Copy from the context menu or by using a CTRL- C keyboard shortcut for Copy.

There are some limitations in this process. For example, a table can be copied and pasted as a drawing only if the table is geocoded or if it contains geometry types. See the Creating Drawings from Geocoded Tables topic.

Note that we can use the Copy, Paste and Paste As buttons either from the project pane toolbar or from the main toolbar. We can also use these commands as they appear in the right-click context menus.

Cut - Copy the highlighted component to the Windows clipboard and then delete it from the project.

Copy - Copy the highlighted component to the Windows clipboard.

Paste - Paste the Windows clipboard into the project.

Paste As - Paste the Windows clipboard into the project as a given component. The down arrow pulls down a menu showing what types of components can be used for the data on the clipboard.

Example: Copy an Image and Paste As a Table

1. Open the project pane.
2. Right-click on the image component to be copied and choose Copy from the context menu.
3. Right-click anywhere in the project pane other than onto an existing component and choose Paste As... and then Table in the context menus that appear.
4. Choose the options desired in the dialog that appears for how the image data should appear in the table. For example, for each pixel we can create separate fields giving the Red, Green and Blue channel values for that pixel.
5. This creates a new table component where each record in the table is taken from one pixel in the image. The table is "geocoded" with the pixel locations of each pixel in the image together with the values of the pixels.

For more sophisticated use, it is also usually possible to Copy from within a component window and to Paste As... the data directly into a new component in the project. For example, we can select points within a drawing and then Copy those points, click into the project pane and use Paste As... a table to create a table that contains geocoded records for each copied point.

Copy and paste works for images and surfaces between Manifold and other applications. For example, one can Copy an image in PhotoShop and Paste it into the project pane as a new image in Manifold or vice versa. Surfaces will be copied as a bitmapped image using whatever display options are currently applied. Images copied from other applications are always pasted as images.

Autoscope

The "scope" of a Copy is what is copied. When copying from an open drawing, image or other window, if a selection is present only those items (image pixels, drawing objects, table records, etc) that are selected will be copied. This is an example of autoscope, where the scope of the copy is automatically restricted to the selection if one is present.
In contrast, when using **Copy** to copy components that are highlighted in the project pane the entire component will be copied whether or not there is a selection within that component.

**Typical Copy and Paste As Combinations**

- **Comment to Script**
  Copies the text of a comment and pastes it as a script.

- **Script to Comment**
  Copies the text of a script and pastes it as a comment.

- **Drawing to Table**
  Reads the coordinate locations of points and pastes them as records into a geocoded table. This is a fast way of creating a list of latitude / longitude locations for points in a map, such as a city.

- **Table to Drawing**
  Paste a geocoded table as a drawing of points.

- **Image to Table**
  Create a table where each record is a "geocoded" pixel from the image.

- **Drawing to Labels**
  Create a labels component with a label in position for each object in the drawing. When copying and pasting between the same Manifold session, can automatically create labels that are populated from the drawing's fields.

- **Labels to Table**
  Create a table where each record is the geocoded location of a text label as well as the text string in the label.

- **Table to Labels**
  Create a labels layer from a table of geocoded records where each record has the location and text string for a text label.

- **Surface to Image**
  Although surfaces are shown in a window as if they were a 2D image, this is just an optional visualization of the data. Surfaces may be copied and pasted as images to provide true images.

- **Terrain to Image**
  Create an image from a terrain where the image pixel values correspond to terrain elevations.

- **Image to Surface**
  Create a surface from an image where pixel values are used to create elevations.

- **Surface to Table**
  Create a geocoded table where each record contains an elevation or other data value for a given point in the surface.

- **Table to Surface**
  Create a surface taking elevation values for point records specified in a geocoded table.

- **Drawing to Surface**
  Create a surface taking elevation values from drawing objects.

**Pasting Drawings as Tables**

When pasting a drawing as a geocoded table, points are represented as a single geocoded record where the latitude and longitude field of the record are taken from the position of the point and the other fields in the table are taken from any data attributes associated with the point. For lines and areas, this position will be the centroid of the object.

The **Paste As Table** dialog allows a choice of which fields will be pasted. By default, all ordinary fields are checked for pasting while **intrinsic fields** are not checked for pasting. Intrinsic fields are fields that are automatically computed, such as **Area (I)**. Intrinsic fields include:

- **Type (I)**
  A lookup field for object type: area, line or point.
Branches (I)  The number of branches (indicates topologically branched objects, such as areas with holes or islands).

Coordinates (I)  The number of coordinates that define the object.

  X (I)  X coordinate of the center of the object in native coordinate system units.

  Y (I)  Y coordinate of the center of the object in native coordinate system units.

Latitude (I)  Latitude of the center of the object in degrees latitude.

Longitude (I)  Longitude of the center of the object in degrees longitude.

Length (I)  Object length (or perimeter).

Area (I)  Object area.

Bearing (I)  0 for points and areas. For lines, the bearing in degrees of a straight line segment constructed between the first and last points of the very first branch within the line. Computed using ellipsoidal trigonometry. 90 is East and 270 is West.

Selection Mask (I)  A byte giving the saved selections mask.

Selection (I)  Boolean: currently selected or not.

The Branches (I) and Coordinates (I) fields will be pasted as Int32 integers, the Type (I) field is pasted as a variable-length Unicode text field and all the other computed fields will be pasted as Float64 floating point numeric fields.

See the Intrinsic Fields topic for more information on intrinsic fields.

**Pasting Images as Tables**

Manifold can paste images as tables where every pixel in the image is pasted as one record in the table.

  Color Index (I)  An index into the palette. Appears as a choice only when pasting palette images.

  Color (I)  A decimal number representing the RGB color. The decimal representation of the six-digit hexadecimal number formed by concatenating the three, two-digit hexadecimal values for R, G and B channels.

  Red (I)  A value from 0 to 255 for the Red channel value.

  Green (I)  A value from 0 to 255 for the Green channel value.

  Blue (I)  A value from 0 to 255 for the Blue channel value.

  Alpha (I)  A value from 0 to 255 for the Alpha channel value.

  Intensity (I)  The intensity of the pixel as a value from 0 to 255.

  Hue (I)  The hue of the pixel as a value from 0 to 255.

  Lightness (I)  The lightness of the pixel as a value from 0 to 255.

  Saturation (I)  The saturation of the pixel as a value from 0 to 255.

  Selection (I)  Boolean: currently selected or not.

  Selection Mask (I)  A byte giving the saved selections mask.
Invisible (I)  Boolean: the pixel is invisible or not.

X Offset (I)  X position of the pixel. The column number of the pixel in image row and column coordinates.

Y Offset (I)  Y position of the pixel. The row number of the pixel in image row and column coordinates.

X (I)  X coordinate of the center of the pixel in native coordinate system units.

Y (I)  Y coordinate of the center of the pixel in native coordinate system units.

Latitude (I)  Latitude of the center of the pixel.

Longitude (I)  Longitude of the center of the pixel.

Color Index (I) appears as a choice only when pasting palette images; however, the RGB and Alpha values appear even if the image is not an RGB or RGBA image. In this case, they represent the R, G, B and Alpha values for the color of the pixel. Note that even though a color may be taken from a palette, it still can be described as composed of R, G and B values.

Pasting Surfaces as Tables

Manifold can paste surfaces as tables where every pixel in the image is pasted as one record in the table.

Height (I)  Height of the pixel.

Selection (I)  Boolean: currently selected or not.

Selection Mask (I)  A byte giving the saved selections mask.

Invisible (I)  Boolean: the pixel is invisible or not. Missing pixels are reported as invisible.

X Offset (I)  X position of the pixel. The column number of the pixel in image row and column coordinates.

Y Offset (I)  Y position of the pixel. The row number of the pixel in image row and column coordinates.

X (I)  X coordinate of the center of the pixel in native coordinate system units.

Y (I)  Y coordinate of the center of the pixel in native coordinate system units.

Latitude (I)  Latitude of the center of the pixel.

Longitude (I)  Longitude of the center of the pixel.

Cut and Paste

Cutting objects from one drawing and pasting them into a different drawing is the default way of moving objects between drawings in Manifold. To avoid the need to think about projections, create your blank drawings by creating one drawing and then copy that blank drawing and paste it as several additional drawings. This will guarantee that all of the drawings use exactly the same projection settings. It's then possible to copy, cut and paste objects between the drawings without worrying about projection nuances.

If you've Pasted something into a drawing and don't see it, try Zoom to Fit to make sure it is not outside the window at the current zoom and pan setting.

Copy and Paste between Applications
Some items within Manifold cannot be copied and pasted to other applications because the other applications
don't understand the objects used within Manifold. For example, Microsoft Word does not understand how to
accept vector objects containing data attributes if objects are copied from a Manifold map and pasted into Word.

Images and surfaces can be copied from Manifold and pasted to any application that understands bitmaps.
Surfaces will be pasted as a bitmap using the current display options.

Another option to transfer images or a view of drawings or maps is to use the Tools - Make Image command to
save to an image component. This captures the desired component as an image. The image can then be used
within Word or other applications.

Copying tables or table columns or records will copy data to the Clipboard both in internal Manifold format as well
as conventional Windows tab-separated text format, which is handy for pasting into Access, Excel and numerous
other Windows programs that use tables.

**Tech Tips**

At times we would like to fetch a component that's been saved in a .map project file for use in another project
file. We can import the component using File - Import - Component directly from the .map file. Another way to
do this is to open a second instance of Manifold and to open the saved .map project. We can then Copy
components from that project file, switch to our current working instance of Manifold and Paste them into our
current project.

Sometimes we would like to transform a component from one form into another, such as converting a surface
into an image, and we don't need to keep a copy of the original component around. In such cases it is faster to
Cut the original component and Paste As a new component. When using Copy Manifold must make a copy of
the data, which can take a long time in the case of very large components.

**See Also**

Don't miss the Copy and Paste: Image / Table / Drawing example for a detailed series of steps using Copy and
Paste As. Starting with an image the example creates a table and then copies that table and creates a drawing.

See the Edit - Paste / Paste Append topic for information on the Paste Objects dialog, which pops up when
copying objects from one drawing and pasting them into another drawing.

See the Download and Mosaic TerraServer Images topic for an example using Copy and Paste to assemble a
larger image out of smaller ones.
Working with Text Components

Comments, queries and scripts are called **text components** because they contain text.

In the case of comments the text contained is whatever we want it to be.

Comment example:

```
Longitude_Resolution: 0.
Geographic_Coordinate_Ur
Geodetic_Model:
  Horizontal_Datum_Name: I
  Ellipsoid_Name: GRS1980
  Semi-major_Axis: 6378137
  Denominator_of_Flattening
```

Comments are often created when importing formats such as **SDTS** that contain "readme" information, metadata or other text information along with GIS data of various sorts.

Query example:

```
SELECT States.* FROM States WHERE
  IsArea(States.ID) AND
  SELECT Max(Area(Counties.ID))
  Where Contained(States.ID, Co
```

Queries contain text that makes up SQL statements.

Script example:

```
Sub SayHello
  str = "Hello!"
MsgBox str
End Sub
Sub Main
  Call SayHello
End Sub
```

Scripts contain text that forms programs written in the scripting language in use.

All three types of windows accept standard Windows text editing shortcuts such as **CTRL-X** to Cut, **CTRL-C** to Copy, **CTRL-V** to Paste and so on. Text component windows also support Edit - Find and Edit - Replace. However, since text components only allow highlighting of one block of text at a time the **Find All** command will be disabled.

All text components support bookmarks, which are icons that can be placed in the left margin to mark a line of text, as seen in the comments component illustrated above. Place a bookmark at any line of text by pressing **F8** or by choosing **Edit - Bookmarks - Toggle** from the main menu. The Edit - Go To command can be used to jump to any bookmark.

To set a bookmark:
1. Open the component.
2. Click on the line where the bookmark is to be placed. This positions the cursor on that line.
3. Press F8 or choose Edit - Bookmarks - Toggle from the main menu.

Script components also support breakpoints, which are icons that mark a line of code at which script execution automatically stops. Breakpoints are used for debugging scripts. Place a breakpoint at any line of text by pressing F9 or by choosing Edit - Breakpoints - Toggle from the main menu. Breakpoints can temporarily disabled to allow code to flow through the breakpoint line without losing the marked position. The script component seen above shows two breakpoints. The Edit - Go To command can be used to jump to any breakpoint.

**To set a breakpoint:**

1. Open the component.
2. Click on the line where the breakpoint is to be placed. This positions the cursor on that line.
3. Press F9 or choose Edit - Breakpoints - Toggle from the main menu.

Bookmarks and breakpoints are most frequently used when writing large, complex scripts.

**Go To**

The Go To command can jump to any line number, bookmark or breakpoint. In the above example we can jump to the breakpoint set at line 7 for the line that contains the text "Call SayHello"

**See Also**

- Edit - Bookmarks
- Edit - Breakpoints
- Edit - Go To
- Queries
- Programming Manifold
Undo / Redo

Many Manifold commands can be undone using the **Undo** button. The keyboard shortcut for Undo is **CTRL-Z**.

Once undone with **Undo**, a command may be redone with **Redo**. The keyboard shortcut for **Redo** is **CTRL-Y**.

The **Undo** command will undo the last issued command. In some cases, Manifold will allow undoing more than one command, but usually it is only the last command that can be undone.

**Very important:** Not all commands and operations in Manifold may be undone. Routine image editing and drawing editing commands, transform operators, selections, copy and paste operations, renaming and other commands can in general be undone. However, some operations with tables and external tables linked into a Manifold project may or may not be undone. Even simple changes to tables if they are linked to an external data provider might not be undoable if the external provider does not support undo. Before making sweeping changes to tables, please take time to make a copy of the table data by exporting it to a file.

**Confirmations**

Some **Undo** actions require making transient copies of data to allow the action to be undone. Certain operations can involve lots of data and thus result in slower performance than desired because of the time required for copying transient data. Manifold provides two options in the Tools - Options - Confirmations page to control this behavior:

- Confirm creating undo actions larger than [50] MBytes
- Never create undo actions larger than [250] MBytes

The sizes used in the above options are user settable, with 50 and 250 MB the respective defaults.

**Note**

The **CTRL-Z** keyboard shortcut for **Undo** is frequently used with editing when operating with the mouse with one hand and commanding cut, copy and paste commands with **CTRL-X**, **CTRL-C** and **CTRL-V** keyboard shortcuts to reverse unintended keyboard commands.
Linked Views

Manifold can automatically pan the view in one window based on where we look in a different window that is the active window, a capability that is managed using the World pane. When a window is set to track the view in the active window we say that the two windows have linked views.

The World pane shows the current center point of the active window on a simplified map of the world. In addition, the World pane is used to control linked views by specifying Show Position Reticule and Track Position settings for individual windows. Open the World pane by choosing View - Panes - World or by using SHIFT-ALT-O to toggle the pane on or off. The World pane may be docked underneath the project pane when using docked panes.

The World pane is context sensitive: it shows the current Show Position Reticule and Track Position settings for the active window. The World pane announces the active window in the title bar and marks the center point of the active window with a reticule. In the example above, we have opened the Mexico sample drawing in a window. The World pane therefore announces Mexico as the active drawing and positions the reticule over Mexico in the World pane's minimap.

Show Position Reticule - Show a reticule in this window that always marks the center position of the currently active window (if it falls within the view seen in this window). Push in to activate.

Track Position - Pan this window so the center position of the currently active window is centered in this window's view. Push in to activate.

(Coordinates Readout) The bottom of the World pane reports the coordinates of the center point of the active window.

To instruct a window to show a reticule at the current position

1. Open a component in a window.
2. Open the World pane and push in Show Position Reticule.

If any other window is now opened (even a second copy of the component opened above) the center point of the current position of the active window will be shown with a reticule.

To instruct a window to track the current position

1. Open a component in a window.
2. Open the World pane and push in Track Position.

If any other window is now opened (even a second copy of the component opened above) the tracking component will be panned so that it always shows the current position of the active window centered in view.
**Example: Displaying Position**

For our first example we will use two drawings showing the same subject. The right-hand drawing will be a zoomed in view. The left-hand drawing will always show where the zoomed-in view is positioned by placing a reticule at that spot.

We begin by importing the sample *Mexico* drawing. We open the *Mexico* drawing in two windows. First we click it open in the window seen on the left and then we right click on the *Mexico* drawing in the project pane and choose **Open in New Window** to open it in another window. We've resized both windows to fit into a screen shot for this documentation and then zoomed into the window on the right. To color the drawing, we use the Color dialog and then change the color using a preset in the thematic formatting dialog to suit our taste.

Note that the focus is on the window on the left (because it has an "active window" title bar.

With the focus on the window on the left, we push in the **Show Position Reticule** button in the World pane.
A reticule appears in the center of the left-hand window. The reticule appears in the center because the active window is the same window.

Suppose we now click on the right-hand window and use **Zoom Box** to zoom into a region in that window.

With the focus on the window on the right, the reticule in the left-hand window will move to the position of the center point of the active window. Because the right-hand window view is zoomed into a location on the East coast of Mexico, the reticule moves to that position.
We can now use a command like Center Point to click in various locations within the right-hand window. As we move to different locations, the reticule in the left-hand window will move to show the current position seen in the right-hand window.

**Example: Tracking Position**

For our second example we will use the same drawings and arrange things so that as we pan the left-hand drawing the right-hand drawing will provide a zoomed-in "moving map" view.

With the focus on the right-hand pane, we press in the Track Position button in the World pane. This commands the window to move so that the center point of the active window is centered in it.

We can now click back to the left-hand window to move the focus there and use Center Point to move about in the window. As we move, the right-hand window automatically pans to follow. In the illustration above we first zoomed into the right hand window to show greater details and then we clicked on the left-hand window.
and zoomed in. We then used Center Point to move around in the left-hand window. If we click on a position on the Northern coast the right-hand window pans to show us that location in greater detail.

![Map Illustration](image1)

If we click Center Point on a spot on the Southern coast the right-hand window again moves to show us that location.

**Example: Linked Views with Maps**

Linked views can be an essential tool for navigating within very dense and detailed maps. In this example we've downloaded CD 155 of the VMAP 1 set from NIMA ([www.nima.mil](http://www.nima.mil)), showing parts of Central and South America at approximately 1:100K-scale paper map equivalent resolution. In the illustrations below, the map on the left shows boundary layers with most layers turned off. The map on the right shows elevation layers, also with most layers turned off. The map on the right is a huge map representing hundreds of megabytes of data (use a fast machine). In particular, unless it is zoomed far into the map the elevation contour lines blend into a black mass.

![Map Illustration](image2)

We click on the left-hand map to make it the active window and then we press Show Position Reticule in the World pane. We then click on the right-hand map to make it the active map and then we press Track Position in the World pane. Now, whenever we move the right-hand map (say, by scrolling with CTRL right clicks of the mouse or by using the Center Point command), the reticule in the left-hand map will show us where our right-hand view is located.
If we click on the left-hand map we can use Center Point to move the center position to any spot and see that spot greatly enlarged in the elevation contour lines map on the right.

**Example: Navigating a Terrain using a Surface as a Guide**

One of the many excellent uses for linked views is to use a surface to steer a view in a terrain. In this example we've imported a surface showing terrain elevation in the Grand Canyon area of the United States. We've opened the surface as a small window and zoomed into the surface. We've also opened the terrain in a window.

To set up the linked views we click on the terrain window and press Track Position. We then click on the surface window and press Show Position Reticule. We don't really need to press Show Position Reticule for the surface window to move the terrain window, but it is usually convenient to have the position reticule appear at the center of the surface window so we can see exactly where the terrain is positioned.
We can now use commands like Center Point to navigate in the surface window and the terrain window will automatically jump to whatever location we center in the surface window. In the views above we've clicked onto the river channel. Note that the terrain window has been rotated a bit to provide a view to the East aligned with the river.

If we click on a different location (again, on the river channel) the terrain jumps to that location as well.
Many GIS programs have some sort of "minimap" navigation pane where one can click into the pane and move the position of the main display to that location. Manifold's linked views capabilities allow us to use any window as a "minimap" control to move any other window.

Newbies will sometimes get mixed up between Show Position Reticule and Track Position. To help keep things straight:

- **Track Position** is a "moving map" command. It tells the window to move to whatever is the current position.
- **Show Position Reticule** shows a reticule at the current position. It does not move the map.

It's slightly redundant, but one can have both Show Position Reticule and Track Position pushed in at the same time. In that case, the reticule shown by Show Position Reticule will always appear at the center of the display.

Do the contents of a window disappear when it has Track Position set? If the active window shows a geographic location that is not covered by the data set in the window, Manifold will still merrily pan the window to a location far beyond the contents of the window. Check to see if the window is correctly georegistered if you think the contents should appear.

More than one window can be set to use Track Position. In that case, all windows with Track Position set will be panned to the new location whenever the active window is panned. Panning lots of windows that contain large amounts of dense data is, of course, computationally intensive and will slow system response.

It's easy to think that linked views work only between two windows. That's not so: linked views settings of Track Position and Show Position Reticule apply to *all* windows that are open and have these buttons set in the World pane.
### About Geocoding

A **geocoded** table is a table where every record has a location given in latitude and longitude, using standard decimal degrees notation for the latitudes and longitudes. Unless each record is geocoded with a latitude and longitude location Manifold cannot know where that record is located. Once a table is geocoded it may be used to create drawings.

<table>
<thead>
<tr>
<th>Towns</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Incline Village</td>
</tr>
<tr>
<td>St. Petersburg</td>
</tr>
<tr>
<td>Lake Oswego</td>
</tr>
<tr>
<td>Lancaster</td>
</tr>
<tr>
<td>Alexandria</td>
</tr>
<tr>
<td>Marquette</td>
</tr>
<tr>
<td>Marblehead</td>
</tr>
<tr>
<td>Oshkosh</td>
</tr>
<tr>
<td>Cheyenne</td>
</tr>
<tr>
<td>Round Rock</td>
</tr>
<tr>
<td>Redmond</td>
</tr>
<tr>
<td>Wichita</td>
</tr>
<tr>
<td>Palo Alto</td>
</tr>
</tbody>
</table>

For example, the table above is not geocoded. It lists the names of towns in the United States but there is no way to tell from the table exactly where the towns are located. If we were to try to draw points on a paper map for each town we would not know where to place the points. If a table is not geocoded, it cannot be used to create a drawing in Manifold either, because Manifold also would not know where to put the points.

<table>
<thead>
<tr>
<th>Towns</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Incline Village</td>
</tr>
<tr>
<td>St. Petersburg</td>
</tr>
<tr>
<td>Lake Oswego</td>
</tr>
<tr>
<td>Lancaster</td>
</tr>
<tr>
<td>Alexandria</td>
</tr>
<tr>
<td>Marquette</td>
</tr>
<tr>
<td>Marblehead</td>
</tr>
<tr>
<td>Oshkosh</td>
</tr>
<tr>
<td>Cheyenne</td>
</tr>
<tr>
<td>Round Rock</td>
</tr>
<tr>
<td>Redmond</td>
</tr>
<tr>
<td>Wichita</td>
</tr>
<tr>
<td>Palo Alto</td>
</tr>
</tbody>
</table>

In contrast the above table is geocoded. Each record now has a latitude and longitude location given in decimal degrees notation. We could use the latitude and longitude values to draw a point for each town on a paper map of the United States. Manifold could also use this table to create a drawing.
If a table is geocoded it can be used to create a drawing, which in turn can be used in a map like the illustration above. Right away, the positions of the towns convey an immediate visual impression of their locations that one does not get in a table presentation. The display of data at the right location is part of the great power of a GIS like Manifold, so we would like our data to be geocoded so the data can immediately be displayed within a map.

The problem is that a lot of the important data sets we deal with, whether they are lists of customer addresses or lists of oil wells or lists of fire hydrants that need maintenance, are not geocoded. The central problem for many GIS users is getting their data geocoded. Depending on the contents of the table, geocoding the table can be a reasonably straightforward process or it can be very difficult or even impossible.

Let's take a look at three geocoding tasks to see what approaches to geocoding are possible in different cases. We will look at geocoding a table of towns, geocoding a table of fire hydrants and geocoding a table of street addresses.

**Geocoding a Table of Towns**

Suppose we have a table with town names like the first example above. How can we geocode it? In the simplest case we look up the latitude and longitude of each town in a reference book or atlas and we add the latitude and longitude to the table for each record by hand. A better way is to use the power of Manifold to automate the process.

If we have a table somewhere that already lists latitudes and longitudes for towns we could extract information from that table automatically. Because there are many geocoded tables of populated places that are easily obtained by free download from the Internet it is usually a straightforward matter to geocode a table of place names.

There are three techniques that are most frequently used:

- If we have another database table that contains the town name and latitude and longitude we could use SQL facilities such as Union to combine the two tables via a relation using a key field such as the name of the town.
- If we have a drawing that shows points for cities in the United States (such as a drawing of populated places) we can geocode the table using the drawing as a guide with Manifold’s spatial Match tool. See the Spatial Geocoding with Match topic for more on this tool.
- If we have the Manifold Geocoding Tools package installed, we can geocode using the town name by following the fast and simple procedures in the Street Address Geocoding topic.

**Geocoding a Table of Fire Hydrants**

Suppose our town would like to create a GIS database of all fire hydrants in the town. We plan to use the power of GIS to help keep track of the status of all fire hydrants and to help plan regular maintenance, cleaning, flushing of water systems and so on. Let’s say we have inherited a database of fire hydrants that provides an identification number for each hydrant, some status information on the hydrant and a “location” field that consist of
a text comment noting what street the hydrant is on and what is the nearest crossing street. Our task is to geocode the table with the latitude and longitude location of each hydrant.

In the United States, the simplest way to accomplish this task is to connect a portable, WAAS-enabled GPS device to a laptop loaded with Manifold, turn on the GPS Console and then drive to the location of each fire hydrant. With Instant Data turned on we would place a point at the location of each fire hydrant and write down the identification number of that hydrant. The result of this process will be a map of points where each point is the location of a single fire hydrant. In addition to the object ID field, the drawing's table will have only one data attribute field in it, the identification number of the hydrant. We can then use this drawing together with Manifold's Match tool to geocode the hydrants database table using the identification number fields as key fields.

Although recording the locations of fire hydrants in an entire town in this way requires a substantial amount of driving, the process goes very rapidly when the GPS Console and Instant Data are used. WAAS-enabled GPS devices can achieve 2-meter (about 6 feet) accuracy, which is sufficient accuracy to locate fire hydrants.

Unfortunately, WAAS is generally available within the United States only. In regions outside the United States GPS devices will provide only 15-meter (about 50 feet) accuracy by default, which many people would not consider sufficient accuracy for mapping fire hydrants. 15-meter accuracy may be fine for locating bridges, which are large structures that are easily found when one is positioned within 15 meters of them, but in the case of smaller objects such as fire hydrants, especially if they are to be placed on digital maps in relation to features such as buried pipes, one normally would like better accuracy.

One way to accomplish this geocoding task outside of the United States would be to drive the city streets and manually mark on a paper map the locations of all hydrants. We could then scan the paper map, georegister the resultant image and then use Tracing to create a drawing of points that show the location of each hydrant. We could then enter the identification number for each hydrant into a data field for each point and then once again use Match to geocode the database table using the identification number as a key field.

Another alternative might be to acquire an aerial photograph of sufficient resolution that hydrants are visible, to scan in the photograph, georegister it and then use tracing to create a drawing of hydrant points and Match to geocode based on the drawing. Although overhead photography is probably not very practical in the case of fire hydrants (which would be obscured by trees in many cities) it is a very practical way of geocoding other infrastructure items, such as bridges or electrical transmission towers. Panning and zooming within a map that contains an image layer and then clicking to create a point at various locations in an overlying drawing is a very fast process.

Note that the task of geocoding a table of fire hydrants is directly analogous to the task of geocoding a table of oil wells, a table of monitoring stations in a forest or, for that matter, any table of items whose location is not known. In all such cases we must determine the latitude and longitude location of each item by either physically measuring the latitude and longitude with a GPS, by marking the location accurately on a map or by determining the locations using an aerial photograph. If the items to be geocoded are easy to reach and a GPS is available the geocoding process might be very straightforward. If they are far away and there is no aerial photograph or other map that can be used, then it could well be impossible to find their locations and thus geocode the table.

Geocoding a Table of Street Addresses

If we had a table of street addresses like the one below we could not plot these on a map because the table is not yet geocoded. Without a latitude and longitude location for each record we would not know where to place it on a map.
The ability to find a latitude and longitude location for a given street address is called street address geocoding. Without a latitude and longitude location for a street address, no GIS package knows where that address really is.

It is easy to make the conceptual mistake of thinking of a street location as being an exactly defined location, the same as a latitude/longitude location. However, that mistake arises mainly from how people use addresses to find locations for the delivery of mail or to go to a particular restaurant or other location. Street addresses, of course, do not really convey an exact latitude and longitude for the address. They simply provide a means by which a postal carrier or someone else physically traversing the streets can find a particular address.

To find an address we have to find the street (with the help of a map if we don't know a particular town), orient ourselves to the address system used on that street and then locate the address. As anyone who has tried to find an out-of-sequence address in an unfamiliar town knows, there is a great difference between hunting down a particular street address and going directly to a latitude/longitude location.

It is one thing to be able to find a given street address by physically going there (perhaps with the help of a local street map) and it is quite another thing to plot a table of street addresses, such as the table of restaurant addresses, on a map as seen above without ever going to the actual address. To plot each restaurant shown in the table we need to know the actual latitude and longitude address at which it is located. To do that, the table must be geocoded as seen below.
In recent years the adoption of geocoding technology by consumer computer applications (at least in many First World nations) has also encouraged us to think of street addresses as being equivalent to a latitude/longitude location for the purpose of computer mapping. Internet mapping sites allow us to enter a street address, such as "525 Main Street, Central City," and instantly see a street map with the location of the address marked as if we had provided an exact latitude and longitude location. Low cost navigation systems that combine GPS technology with built in maps and street address geocoding systems allow us to specify a street address and navigate directly to that location, again, as if we had given exact latitude/longitude coordinates for our desired destination.

As a result, it is quite common for people to expect to be able to enter a street address into a web site or a map and to see a physical location for that address, a sort of "geocoding on the fly." Some applications may give the appearance of taking a list of addresses and displaying them straightaway as points in a map; however, in all cases the software will internally take the intermediate step of using the address to determine a latitude and longitude location for the record. The latitude and longitude location is then used to plot the location of the point.

Software packages use many different strategies to geocode street addresses into latitude and longitude locations. The basic approach is to maintain a large database of streets and address ranges so that the location of a particular address can be estimated from the database. Software that can perform street address geocoding may be built into a GIS package, it may be sold as separate geocoding software, or it may be provided as an Internet web service.

If we install the Manifold Geocoding Tools package, we can turn on the street address geocoding capability that is built into Manifold. The Manifold street address geocoding engine becomes functional when the Geocoding Tools package is installed and a geocoding data source is available, such as the Manifold Geocoding Database for US streets that is provided on the Manifold downloads site. If we have installed Geocoding Tools and a geocoding data source we can use Manifold to geocode a table of valid US street addresses with the approximate position of each address.

To geocode street addresses outside the US, other geocoding data sources may be used. For example, Microsoft’s MapPoint product may be used as a geocoding data source for addresses in Canada or in various European countries. See the Geocoding Data Sources topic.

**How Street Address Geocoding Works**

Any geocoding software (including Manifold) that provides street address geocoding must find the street address and an equivalent location in a database. Unfortunately, there are no global databases that provide an accurate location for each street address although it is possible through special means to create local databases that have accurate locations for street addresses.

To take the United States for example, there is no national database that specifies exactly where all addresses are located. This is mainly because addresses in the US are highly irregular, are poorly documented and change too rapidly for either private companies or government agencies to be able to keep up with perfect accuracy. Therefore, when using any street address geocoder it is important to understand that the output of the geocoder is almost certain an approximate location and not an exact location.

The closest approximations to a national database of address locations that exist in the United States are the U.S. Bureau of the Census TIGER database and the TIGER/Line data sets derived from TIGER. TIGER/Line attempts to show known roads with address ranges for each road segment. Actual addresses are not noted, but...
are represented only as a best effort at showing the address range (from lowest to highest address number) that occurs in a particular street segment. Most geocoding software in the United States, including Manifold, uses databases that are derived in some way from the TIGER/Line data sets.

Based on data sets like those created by the Census Bureau, geocoding software can be created that compares a record's address, such as 525 Main Street, Central City, Idaho, 01120 to an internal database of street segment coordinates and address ranges for each segment. For example, after zeroing in to Main Street by using State, ZIP code and City fields, the software can find the right Main Street segment that contains the address range for the address number at hand. If one particular Main Street segment has a high value of 600 and a low value of 500 for the address range on that segment, the software could then reasonably infer that 525 Main Street is located about one fourth of the way up that particular street segment. It could then assign the latitude and longitude of that interpolated spot to the record.

It is important to understand that the geocoding software has no idea where the actual address is located. It simply interpolates the location of the address by making what is hopefully a reasonable guess based on the address range recorded for a given street segment. Clever software can use a variety of strategies to make better guesses, but at the end of the day the results are usually accurate to only within a city block in urban areas and are wildly inaccurate in rural areas. Addresses of the form Rural Route 10 Box 82, for example, in a rural area might not be geocodable to within tens of miles if they are geocodable at all.

Note that the above method for creating and using geocoding databases does not store the actual locations of specific street addresses. It only says that if there are any street addresses on, say, this particular street segment of Main Street they would fall into the range between 100 and 200 Main Street. If we ask the system to geocode 150 Main Street the system doesn't question our implied assertion that there really is an address of that nature, it simply takes it on faith that if we are asking to geocode such an address we want the system to report what the computed location should be and so the system plops a point half way up that segment of Main Street.

The geocoding database does not actually know there are any addresses on that segment of Main Street or, if there are any, what those addresses might be. It simply knows that if there are any addresses on that segment of Main Street those addresses will be between 100 and 200. As far as the database is concerned there could be empty fields on that segment. Because the database is structured in this way it is not possible to reverse geocode using segment and address range data.

Reverse geocoding is the process of finding all addresses that are within a given range of a given location. Reverse geocoding can only be accomplished with any sort of accuracy if the database in use is a points of interest style database that contains not a database of segments with an address range on each segment but rather contains a database of points with the location of each point and the specific address for each point. Such databases also allow precise accuracy for each address instead of an estimated, interpolated location for geocoded addresses.

Creating geocoding software that can accurately assign an exact, non-interpolated location for each individual address requires a database of all addresses and their exact latitude and longitude locations, that is, the points of interest style of geocoding database. To support 911 service and other emergency response services, some towns are using GPS equipment to create precise databases that show the exact location of each address in their town. Manifold can use such user-provided data sets for precise address geocoding and for reverse geocoding. See the Geocoding Data Extensions topic.

For information on using Manifold's street address geocoding system, see the Street Address Geocoding topic.

Street Address Geocoding Outside the United States

Unfortunately, the United States is the only country that places large government databases of street address ranges like TIGER/Line into the public domain. In other countries, acquiring a database that shows streets and address ranges for those streets is very costly, and in many cases not possible.

As a result, there are many fewer choices for street address geocoding software outside of the United States. Because the Manifold Geocoding Database is based on public domain government data, it is possible for Manifold to provide it for no additional cost on the Manifold download site. Because there is no public domain data for streets and address ranges outside the United States, Manifold provides no geocoding databases for locations outside the United States. However, Manifold includes an option to use Microsoft's MapPoint product as a geocoding data source for addresses in Canada or in eleven European countries. See the Geocoding with MapPoint topic.

Another way to do geocoding outside of the US is to create our own point location geocoding database and then use that as a geocoding data source. A fast way of creating such a point location geocoding database is to use Manifold's GPS Console with a portable computer. Two people ride in a vehicle with one person driving and the
other person operating the computer. They drive to each address and the computer operator uses Instant Data to add a point at that location with the exact address. This can be a very fast process, so fast that an experienced, determined team of two people in a single vehicle can geocode thousands of addresses per day. A town of 50,000 addresses can usually be geocoded in well under a week’s time.

The resultant point location database can then be used for geocoding with Manifold. See the Geocoding Data Extensions topic.

**Geocoded Tables use Decimal Degree Notation**

Geocoded tables in Manifold must have valid latitude and longitude fields consisting of degrees from 0 to +/- 180 longitude and 0 to +/- 90 latitude, with partial degrees denoted as decimal fractions. A minus sign denotes West longitudes and South latitudes. This style of writing latitudes and longitudes is called **decimal degrees**.

Like all modern GIS packages, Manifold uses decimal degree notation because it is an unambiguous standard that is well suited for arithmetic operations and can be written in database tables as text fields or numeric fields. Older methods of writing latitudes and longitudes, such as the use of the letters “E”, “W”, “N” and “S” or the use of degrees, minutes and seconds notation are not well standardized and involve clumsy notation that is not very useful in computing operations. Manipulating values such as *East 32 42’ 15”* is somewhat akin to trying to do longhand multiplication using Roman numerals… not very efficient or sensible.

In modern times most databases of geocoded information use decimal degrees. However, over the years there have been many different styles used to write latitude and longitudes in database tables. Older tables might use text fields to express coordinates in the form of degrees, minutes and seconds, for example. Other tables may use degrees, minutes and decimal fractions of degrees. Some tables will denote longitudes in degrees from 0 to 360. Others might use text strings and prefix a letter, such as “N”, “S”, “E” and “W” to indicate North or South latitudes and East or West longitudes.

Manifold’s approach to dealing with such tables is to import them into Manifold where Transform toolbar operators and other tools can be used to convert coordinates into standard decimal degree notation. This allows the full power of Manifold tools to be brought to bear to adjust the table into the desired form. Clever use of token operations will allow transformation of any format into the desired decimal degrees. See, for example, the Using Tokens and Text Strings topic and the Extract Last Names using Tokens example.

If you have a table that has latitude and longitude values using some old-fashioned notation you should first translate those values into modern decimal degree notation. Only then is it safe to consider it a geocoded table.

**Note:** Although this documentation is written to require decimal degrees notation, in fact if coordinate columns in a geocoded table use degree - minute - seconds notation Manifold will try to parse the notation used to extract valid longitude and latitude values. manifold.net strongly recommends using decimal notation to avoid any possible ambiguity.

**“Generic” Geocoding Strategies**

Geocoding a table by specific addresses is often not required. Although it is easy to understand the conceptual appeal of adding an exact latitude and longitude position for each customer record by address, such geocoded tables also lay a conceptual trap for the unwary in that they are intrinsically inaccurate. Sometimes it is better to have an approximate table that does not lay claims to false accuracy. For many GIS purposes it may be enough to simply pin down a customer location to a specific ZIP code and not to a specific city block. This can be done using spatial geocoding with Manifold’s **match** command. See the Spatial Geocoding with Match topic for spatial geocoding within Manifold.

By spatially geocoding tables using key fields we can often end up with a geocoded table that combines our desired records with a latitude and longitude position for each record. The classic example is displaying customer address records using their ZIP codes. If we have a drawing that shows a point for each ZIP code centroid we can merge the customer address table into this drawing using the ZIP code as a key field. In that case, customer records will appear as a point at the ZIP code centroid for their ZIP code. [The Manifold street address geocoding engine can also geocode addresses that consist only of ZIP codes as well, by geocoding the address to the ZIP code centroid, so as a practical matter there is no need to use Match to geocode to ZIP codes in the US if we have the Geocoding Tools package installed.]

This “generic” method of geocoding is often the only possible method of geocoding for international users who do not have access to a street address geocoding data source for their locations but who do have postal code or telephone code maps or other data sets that can be used as guides for spatial geocoding based on Match.
GIS Jargon and the “Geocoding” Word

New GIS users are sometimes worried about being heard using words in an atypical way that betrays their inexperience, so let's take a moment to discuss a fine point of nomenclature.

The word geocoding as used in this topic and generally used in the GIS world is a jargon or slang word that is reserved for the specific case of adding latitude and longitude values to records in a table in order to specify the geographic location of each record. People who are new to GIS or people who use their jargon in sloppy ways will at times use the word geocoding to refer to the general way GIS drawings keep track of the locations of things. That's a mistake that denies us the usefulness of having a handy slang word that means a single, precise thing.

GIS drawings contain geometry information that describes the shape and location of the objects they contain by recording the coordinates that define the shape and location of the objects. The geometry information is called the object metric (another handy slang word). It's true that in the case of single points the coordinates that define the location of a point are indeed a single pair of latitude/longitude values just like those that might appear in a geocoded table giving a list of records that are to appear as points. However, in the case of lines or areas there are many coordinates that define those objects. We don't therefore refer to the geometry of objects in drawings as being “geocoded.”

Therefore, while a verbally-imprecise or inexpert user might say things like "I'm going to geocode those farm boundaries by tracing over an aerial photo," that's not really using the word "geocode" as it is normally expected to be used. One might say, "I'm going to geocode that table of addresses" but one would not normally talk about "geocoding" the shape of an area. Instead, one might say things like "I'm going to digitize those boundaries from a photo" or "I'm going to trace those boundaries..." or even perhaps (awkwardly) "I'm going to locate those boundaries."

Usually in drawings once the drawing is created if it needs to be cast into the correct geographic context we speak in terms of georegistering or georeferencing it (synonyms). The more subtle idea here is that it is not so much that we have chains of coordinates that correctly define the object metric but that we care about the important nuances of placing those coordinates correctly within an Earth coordinate system such as a projection and a specific location.

Our example of geocoding a table of fire hydrant locations brings us right to the edge of accepted use of the geocode word. If we speak in terms of adding latitude / longitude values to that table of locations then we can be righteous about using the term geocode to describe what we are doing. On the other hand, suppose we have a digital map with a layer of streets and we are creating points in the drawing where the hydrants are located: in that case, we would talk in terms of locating the hydrants on the map or digitizing hydrant locations. So we might say something like "I've got to digitize these hydrant locations from that scanned image so I can geocode that table."

Geocoding is normally something that is done with tables which are then used to create drawings, or something that is done with individual records so they can then be displayed in drawings. For example, a web site designer might say, "When the visitor provides a street address I need to geocode it so I can figure out what part of the map to show."

Coordinate Columns

The latitude and longitude columns used in geocoded tables are also called coordinate columns, because they give the latitude and longitude coordinates at which each record is to be located. A more general form of geocoded tables occurs if the columns are labeled X and Y or have some other column names and give coordinates other than simple latitude and longitude values.

For example, a geocoded table might have coordinate columns that give the actual X and Y coordinates that locate the record in some specific projection, that is, if some specific coordinate system other than simple Latitude / Longitude projection. We're getting ahead of the game for those readers who are going through this documentation in a straight progression of topics, because we have not yet read any of the projections topics. But for the time being, trust us on this: we can have coordinate columns other than simple latitude or longitude which give the X and Y values in some projection for each record. If so, that table also is a geocoded table.

Geocoded Tables and Geometry Columns

Geocoded tables are conceptually the easiest way for new users to understand how a table can store items that are to be displayed in a drawing. It's easy to think about: each record has a latitude and longitude location saved in some coordinate columns. Place a point at that location. The simplicity of this idea is why we introduce
geocoded tables in the Introduction chapter. It is also why most GIS products have some way of working with geocoded tables.

But Manifold can do more, much more. A more sophisticated, more powerful and ultimately higher performance way of saving items within tables is to use geometry columns, which actually save the object metric in a column using some appropriately clever technology. Tables that can save the full object metric can have each record represent not just points but also lines or areas as well, no problem.

There are many possible ways of doing this, but there are surprisingly few GIS systems that can do it in a sophisticated way or even do it at all. In fact, there are so few systems that can do it at all that there has not even evolved a standardized slang term for such tables. Everyone in GIS knows what a geocoded table is, but for those tables that use geometry columns, well, what do you call them? There is no standard term.

At manifold.net we refer to such tables as geometry tables to make clear we are talking about tables that have one or more columns containing true geometry data as opposed to the very simple case of geocoded tables. We use the term geometry table regardless of the specific technology used to implement the geometry column. For example, we use the term geometry table to describe a table that stores geometry using OGC WKB “well known binary” form as well as to describe an Oracle Spatial table storing geometry using Oracle’s high performance SDO_GEOMETRY data type.

Why should we bother to give such tables their own special name?

The reason is that the GIS world tends to divide into two classes of users and applications: those users who do relatively simple, straightforward things using geocoded tables and would like to do so as simply and in as straightforward a manner as possible, and those users who need to operate at a quantum leap higher in sophistication, typically higher end enterprise users, who do far more complex things using geometry tables and who want the full power of tools designed to work with such data.

To support users doing more simple and straightforward things Manifold provides a set of simplified dialogs and capabilities that let us work with geocoded tables in pretty much the same fashion that any simple GIS can do. These features make it convenient and easy to use geocoded tables in a wide range of applications, especially when we need to interface with our colleagues from the database world or web site design world who really don’t want to learn anything about GIS. This documentation includes topics that are dedicated to using geocoded tables without getting into more complex things.

In contrast, those users working with geometry tables can work to their heart’s content and maximum level of expertise doing all the wonderful things made possible by geometry in tables. Those users who don’t need the higher level of function provided by geometry tables can stick to using geocoded tables. We introduce the notion of geometry tables here so that users teleporting into other topics are not confused when they read about geometry in tables in addition to the simple case of geocoded tables.

Notes

The table of sushi restaurants lists sushi restaurants approximately within one mile (1.6 km) of the main USGS facility in Menlo Park. The map in the illustration shows the USGS facility as a yellow diamond and plots the restaurants as green dots. Once we get over our astonishment at the provinciality of a region that can support no more than eleven sushi restaurants per mile, we can see that the restaurants in Palo Alto in the lower right are more tightly clustered than those in Menlo Park, which tend to be spread out along a single main road.

The illustration should not be used for navigation (none of the illustrations in Help should be used for navigation!) since the nature of sushi restaurants is to come and go almost like the seasons. Even at the time of publication of this document several of the restaurants mentioned have disappeared and have been replaced by others.

See Also

Creating Drawings from Geocoded Tables
Geocoding
Geocoding Tools
Street Address Geocoding
Geocoding Data Sources
Geocoding with MapPoint
Geocoding Data Extensions
Manifold Geocoding Servers
Spatial Geocoding with Match
Create a Linked Drawing from a Geocoded Table
Create a Map from a Geocoded Table
Creating Drawings from Geocoded Tables

A **geocoded** table is a table where every record has a location given in latitude and longitude. In Manifold, it is easy to create a drawing from a geocoded table. We simply copy the table and paste it as a drawing.

For an example of a geocoded table consider the following table:

<table>
<thead>
<tr>
<th>Longitude</th>
<th>Latitude</th>
<th>Name</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>-119.47</td>
<td>39.02</td>
<td>Incline Village</td>
<td>NV</td>
</tr>
<tr>
<td>-82.68</td>
<td>27.56</td>
<td>St. Petersburg</td>
<td>FL</td>
</tr>
<tr>
<td>-123.42</td>
<td>46.2</td>
<td>Lake Oswego</td>
<td>OR</td>
</tr>
<tr>
<td>-75.25</td>
<td>40.17</td>
<td>Lancaster</td>
<td>PA</td>
</tr>
<tr>
<td>-77.05</td>
<td>38.91</td>
<td>Alexandria</td>
<td>VA</td>
</tr>
<tr>
<td>-74.34</td>
<td>39.49</td>
<td>Margate</td>
<td>NJ</td>
</tr>
<tr>
<td>-70.84</td>
<td>42.5</td>
<td>Marblehead</td>
<td>MA</td>
</tr>
<tr>
<td>-88.55</td>
<td>44.02</td>
<td>Oshkosh</td>
<td>WI</td>
</tr>
<tr>
<td>-104.79</td>
<td>41.15</td>
<td>Cheyenne</td>
<td>WY</td>
</tr>
<tr>
<td>-97.67</td>
<td>30.52</td>
<td>Round Rock</td>
<td>TX</td>
</tr>
<tr>
<td>-122.11</td>
<td>47.67</td>
<td>Redmond</td>
<td>WA</td>
</tr>
<tr>
<td>-97.04</td>
<td>37.68</td>
<td>Wichita</td>
<td>KS</td>
</tr>
<tr>
<td>-122.15</td>
<td>37.45</td>
<td>Palo Alto</td>
<td>CA</td>
</tr>
</tbody>
</table>

Each record has a value in the latitude and longitude columns. The latitude and longitude of each record specify where that record is located. Note that the latitude and longitude values should be expressed in decimal degrees, the standard GIS notation for latitudes and longitudes. For an introduction to geocoding, see the About Geocoding topic.

Creating a Drawing from a Geocoded Table

Geocoded tables may be used to create drawings. Drawings created from geocoded tables contain points at the latitude and longitude positions of each record in the table.

The illustration shows a map where a drawing of points created from the geocoded table is overlaid in a layer above a drawing showing the United States. The points for each town were plotted at the latitude and longitude of that town as given in the table.
There are three ways to create a drawing from a geocoded table. The first method is to copy the table and paste it as a drawing. The second method is to import a drawing directly from an external geocoded table. The third method is to create a linked drawing from a geocoded table that is either part of the project or residing in some file or database external to Manifold.

To create a drawing from a geocoded table:

1. Import the table into a Manifold table.
2. In the project pane, use **Edit - Copy** to copy the table and then use **Edit - Paste As - Drawing** to paste it as a drawing. A new drawing will appear in the project pane.

To import a drawing from an external geocoded table:

1. Choose **File - Import - Drawing** and open the desired database file or connect to the desired data source.
2. Choose the table desired and check the desired fields. Specify the fields that contain latitude and longitude values (by default, the system will guess that any fields named "Latitude" and "Longitude" are the fields to use). Press **OK**. A new drawing will appear in the project pane.

### Pasting New Drawings and Copying Fields

Pasting a geocoded table as a drawing in the project pane will create a new drawing together with a new associated table in the project. The drawing will contain points at the location of each record.

The associated table (like the tables for all drawings) will be created with at least one field, the object ID field. During the paste process a **Paste As Drawing** dialog will appear asking if any additional fields should be pasted from the geocoded table into the new drawing table. If the **Latitude / longitude coordinates** check box in the **Paste As Drawing** dialog is checked, drawings will be created in Latitude / Longitude unprojected form.

For example, if copying the geocoded table illustrated above we might wish to paste the table into a new drawing as points without copying any of the data fields. In that case, the new table created with the drawing will be empty except for the object ID field. In another example, we might wish to copy the **Name** field as well so that each point record has an object ID field and a **Name** field containing the name of the town at that point. Depending on our desired use of the data we might import all of the fields from the original table. We might even wish to import the latitude and longitude fields as well.

**Very Important:** If we copy the latitude and longitude fields from the original table into the table associated with the drawing, it is very important to realize that these are just copies of the latitude and longitude fields that were part of the original table's records. They are not a "handle" into the geometric coordinates that define the position of the points within the drawing's internal structure. If we click open the table and change these values they will not move the points in the drawing about. See the Editing Data in Tables topic for some additional notes on this concept.

### Geocoded Data and Projected Coordinates

This section involves more advanced concepts requiring an understanding of projections. So far this topic has discussed geocoded tables where each record has valid latitude and longitude fields in decimal degrees. Geocoded tables can also have locations for each record expressed in projected coordinate measures within a certain projection. Such geocoded tables can be copied and pasted to create a new drawing, imported to create a new drawing or used to create a linked drawing.

When using a table that is geocoded using projected coordinates as a new drawing into the project pane we must uncheck the **Latitude / Longitude coordinates** box in the import or pasting dialog. Manifold will then create the new drawing using default Orthographic projection values. Once the drawing is created we can open it and change the drawing's coordinate properties into whatever projection coordinate system was being used by the coordinate values in the table. To do this, we use the **Edit - Assign Projection** dialog after the import to make any necessary adjustments. See the Import a Shapefile example topic for a simple example and the Import Drawing - SHP, Shapefiles topic for a more extensive example of using **Edit - Assign Projection** to adjust projection parameters after import.

### To paste a table geocoded with UTM coordinates
1. Import the table.
2. Copy the table and paste as a drawing.
3. In the Paste As Drawing dialog, uncheck the Latitude / longitude coordinates box.
4. Choose the fields giving UTM coordinates for X / longitude and Y / latitude and press OK.
5. Open the drawing created by the Paste As operation.
7. In the projection dialog, choose the UTM zone that was used for the table's coordinates and press OK.

Geocoded Tables Define Points by Default

When we consider a geocoded table like the examples above it is pretty obvious how each record in such a table corresponds to a single point. Each record has a single latitude/longitude location, so it makes sense to create a point at that table. GIS newbies will often ask, "Why not have a geocoded table that defines lines and areas too?"

That's actually not a bad idea and, in fact, is accomplished at a more expert level by using geometry tables. However, usually the request is to store lines in a table using coordinate columns that give lists of latitude and longitude locations, just as a geocoded table gives locations for points.

The problem that there is no obvious, single standard for defining what records in a table represent a line or an area. A line, for example, consists of a sequence of locations, each of which must be specified. What would such a table look like? Would there be one record that gives the beginning location of a line followed by a series of records that give each location in the body of the line until the last one? Where would the data attributes go? Would they be repeated for each record that gives one of the locations in a line? How would the table indicate that some records define points, some define locations in a line and some define locations in an area? How are multi-branched objects, like areas with holes or islands in them, represented?

It's fairly easy to work out a simple standard for representing points, lines and areas in tables. The problem is that everyone who has done so seems to have worked out a different standard. As a result, there is no single recognized way of encoding anything other than points within geocoded tables. Instead, people use various GIS formats, such as shapefiles or mid/mif files or other typical formats to save and exchange GIS information that contains lines and areas as well as points. If they want to use tables to exchange such information, they use a more sophisticated approach of using geometry tables, which have a "geometry" column that can for each record define the metric of areas, lines or points.

Manifold does allow one exception to the above, a method of saving coordinates in geocoded tables that allows lines to be created when a drawing is linked to that table. Although there is no standard GIS method for representing lines in geocoded tables, Manifold uses a simple convention that can be used to create lines in linked drawings from tables.

When importing or linking a drawing from a geocoded table, we may optionally specify a LineID column that will be used to identify which points will be used to draw lines. However, note that this convention applies only when creating linked drawings or importing drawings from geocoded tables and does not apply when copying a table and pasting it as a drawing. In the copy / paste operation only points are created.

Tech Tip: Changing Ordinary Fields in Tables does not Move Objects

This is a challenging idea for newcomers to GIS who may find it difficult to keep in mind the distinction between the original geocoded table used to create a drawing and the drawing's table.

Recall that every drawing in Manifold has its own table associated with the drawing. The drawing's table contains the data attributes (if any) for the objects in the drawing. If there are no data attributes, then the drawing's table will simply contain one column, the object ID, for each object.

The positions of objects in drawings are embedded within the geometry of the drawing without need to have latitude and longitude values listed in a table. Sometimes we create drawings from geocoded tables that have a latitude and longitude for each record. We might wish to have the latitudes and longitudes that were used to position the points immediately at hand in the drawing's table and so when pasting the geocoded table as a drawing we may tell Manifold to bring along the latitude and longitude fields along with the other fields.

Whatever the motive or process that was used to create a drawing, it is often the case that "latitude" and
"longitude" numbers will be copied to fields in a table as well as being embedded within the geometry of a drawing. GIS beginners often mistakenly think that the positions of items in the drawing are determined from the listings of latitude or longitude fields in the table, so that editing these fields will move the objects. That's not the case.

We can use intrinsic fields to display in a table the actual latitudes and longitudes of objects associated with records. However, intrinsic fields are special, system-generated fields and are different than "latitude" and "longitude" fields we might bring into a table that are ordinary text or numeric fields. The rest of this topic explains why this is so.

Drawings are often created in Manifold from geocoded tables. A geocoded table is a table where each record has latitude and longitude fields that specify the location of that record.

Suppose we have an Access database that contains a list of towns together with their latitude and longitude coordinates:

<table>
<thead>
<tr>
<th>Towns: Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>Longitude</td>
</tr>
<tr>
<td>-----------</td>
</tr>
<tr>
<td>-119.47</td>
</tr>
<tr>
<td>-82.56</td>
</tr>
<tr>
<td>-123.42</td>
</tr>
<tr>
<td>-75.25</td>
</tr>
<tr>
<td>-77.06</td>
</tr>
<tr>
<td>-74.34</td>
</tr>
<tr>
<td>-70.34</td>
</tr>
<tr>
<td>-88.56</td>
</tr>
<tr>
<td>-104.79</td>
</tr>
<tr>
<td>-97.67</td>
</tr>
<tr>
<td>-122.11</td>
</tr>
<tr>
<td>-97.34</td>
</tr>
<tr>
<td>-122.15</td>
</tr>
</tbody>
</table>

We can import this Access table into a table in Manifold [either by importing the data or by linking to the external table].

216
To create a drawing from this table we use **Edit - Copy** to copy the table and then we use **Edit - Paste As - Drawing** in the project pane to create a new drawing based on the data in the table. Each geocoded record will appear as a point in the drawing.

<table>
<thead>
<tr>
<th>Longitude</th>
<th>Latitude</th>
<th>Name</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>-119.47</td>
<td>39.02</td>
<td>Incline Village</td>
<td>NV</td>
</tr>
<tr>
<td>-82.68</td>
<td>27.56</td>
<td>St. Petersburg</td>
<td>FL</td>
</tr>
<tr>
<td>-123.42</td>
<td>46.2</td>
<td>Lake Oswego</td>
<td>OR</td>
</tr>
<tr>
<td>-75.25</td>
<td>40.17</td>
<td>Lancaster</td>
<td>PA</td>
</tr>
<tr>
<td>-77.05</td>
<td>38.91</td>
<td>Alexandria</td>
<td>WA</td>
</tr>
<tr>
<td>-74.34</td>
<td>39.49</td>
<td>Moraga</td>
<td>NJ</td>
</tr>
<tr>
<td>-70.64</td>
<td>42.5</td>
<td>Marblehead</td>
<td>MA</td>
</tr>
<tr>
<td>-88.55</td>
<td>44.02</td>
<td>Oshkosh</td>
<td>WI</td>
</tr>
<tr>
<td>-104.79</td>
<td>41.15</td>
<td>Cheyenne</td>
<td>WY</td>
</tr>
<tr>
<td>-97.67</td>
<td>30.52</td>
<td>Round Rock</td>
<td>TX</td>
</tr>
<tr>
<td>-122.11</td>
<td>47.67</td>
<td>Redmond</td>
<td>WA</td>
</tr>
<tr>
<td>-97.34</td>
<td>37.69</td>
<td>Wichita</td>
<td>KS</td>
</tr>
<tr>
<td>-122.15</td>
<td>37.45</td>
<td>Palo Alto</td>
<td>CA</td>
</tr>
</tbody>
</table>

When we use the **Edit - Paste As - Drawing** command to create a new drawing, Manifold will automatically create the associated data attribute table for that drawing. By default, that new table will have at least one field, the object ID field. During the **Paste As** operation Manifold will give us the option of copying additional fields from the original table into the new data attribute table for the drawing. The dialog presents a list of check boxes for each field in the original table. We can copy any of the fields we desire.

For example, if our geocoded table is a list of customer records containing fields such as name, address, city, state, and so forth together with a latitude and longitude for each record, we might wish to copy the name, address and other informational fields. There is no particular technical reason to copy the latitude and longitude fields (these fields in the original table will be used automatically to position the points in the drawing for each record) but we can do so if for some reason we want to have this information in the table.

If we do copy the latitude and longitude fields into a table associated with the drawing, we will see them in the table window when the table is opened. It is very important to realize that these are just table fields that were copied from the original table. They are not a “handle” into the geometric coordinates that define the position of the points within the drawing’s internal structure. Altering the values will not move the points, nor are the fields even necessary to keep the points “positioned” at the right spots.
For example, in the Table window we could delete the "Longitude" field. We could delete the "Latitude" field as well. If we do so the points do not magically disappear from the drawing. They will still be there. All that we have done is delete a field from the data table associated with the drawing. The points in the drawing have already been created when Manifold read the original database. Changing the text data in the table (either the original one or the new table created with the drawing) changes nothing about those points and does not alter the coordinate numbers inside the drawing.

The above discussion may seem obvious to experienced GIS users; however, for new users it is easy to look at a drawing's table that is full of records with latitude and longitudes values for each and to forget that changing the values in the table does not actually move the points about.

How then can we change the actual location of points in a drawing? There are four main ways:

- We can show the Latitude (I) and Longitude (I) intrinsic fields and then change their values. See Editing Intrinsic Fields in Tables. This is the best method when points must be moved to given coordinates.
- We can right click on individual points in the drawing and use the Object - Coordinates dialog to change the location of the geometric coordinates that define the points. This is a seriously tedious approach.
- We can use any one of a wide variety of interactive editing tools in a map or drawing window to move the points by selecting them and dragging them to a new location.
- If the drawing is a linked drawing created from a geocoded table we can directly edit the latitude and longitude values in the external table and then use View - Refresh Data to update the drawing and move the points.
Note that neither of the above options will change the contents of any "latitude" or "longitude" text fields we might see in a drawing's table.

Remember that this discussion so far is about latitude/longitude values that might be copied into a drawing's table when that table is created. The situation with a table that controls a linked drawing created from a geocoded table is different. Linked drawings created from geocoded tables are drawings of points that are automatically created from and controlled by an external geocoded table using the latitude and longitude values in the table for each record. In addition to the external table that controls them, linked drawings (like all drawings) have their own tables as well.

Attempting to change **Latitude (I)** and **Longitude (I)** values in a linked drawing's own table won't move the points about in the drawing because these fields are read-only when a linked drawing is created from a geocoded table (the other, non-intrinsic fields usually will be read/write). However, changing values in the external table that controls the linked drawing and doing a **Refresh Data** will move points about.

A final subtlety: keep in mind that a linked drawing can be created from a geometry column in external tables. In such cases, the **Latitude (I)** and **Longitude (I)** values in the drawing's table will usually be read/write and may be edited to move the points about.

**Note**

The **Skip zero latitude / longitude records** option is checked by default in the **Paste As Drawing** dialog. If checked, any records in the table that have a zero value in the latitude and longitude fields will not have an object created in the drawing. This is a safety measure so that when pasting geocoded tables that have zero values for latitude and longitude (and thus, are not really geocoded) spurious points will not be created for the zero valued records.

**Creating Linked Drawings from Geocoded Tables**

A geocoded table is a table where every record includes latitude and longitude values in decimal degrees notation that give the position of that record. A **geocoded linked drawing** is a drawing that is created from the data in a geocoded table and intermediate query. Geocoded linked drawings are read-only drawings that by default contain points and optionally may be configured to include lines as well. The location of each point in a geocoded linked drawing is taken from the geocoded table via an intermediate query.

**To create a linked drawing from a geocoded table:**

1. Choose **File - Link - Drawing** and open the desired file or connect to the desired data source.
2. Choose the table desired and check the desired fields. Specify the fields that contain latitude and longitude values (by default, the system will guess that any fields named "Latitude" and "Longitude" are the fields to use. Press OK.

Four new components will appear in the project:

- **A linked table** - This linked table is a way of linking the table data from the geocoded table that will provide the data for the linked drawing.
- **A query** - This query transforms the ordinary, geocoded data in the coordinate columns into geometry data from which the linked drawing will be created.
- **A linked drawing and the linked drawing's table** - The linked drawing is like any drawing, except it is created dynamically from the data within the geocoded table.

Why the intermediate query? It is there to make life easier for inexperienced or casual users - the query makes it possible for Manifold to use very powerful, general purpose spatial technology for linked drawings while still retaining a simplified ability for less experienced or more casual users to easily create linked drawings from geocoded tables.
Geocoded tables are a great convenience for simple applications, but for big-time GIS work they are too limiting. A more powerful way of working is to use **geometry tables**, which save information about the objects they define in a geometry column. In fact, this is such a more powerful technology that all Manifold linked drawings are created from geometry data.

However, to save users from having to create geometry tables when they want to do something so simple it can be accomplished with geocoded tables, Manifold has special internal capabilities to allow us to create linked drawings from geocoded tables. When we attempt to create a linked drawing from a geocoded table, Manifold can tell what we are using is not a geometry table. In that case, it automatically writes a query that converts the geocoded table data on-the-fly into geometry data and then links the drawing from that query.

This retains the simplicity of a geocoded table while allowing all linked drawings to use a common internal technical framework. The chain of action from the original geocoded table into the geocoded linked table in the project through the query into the linked drawing is perfectly dynamic: changing a latitude or longitude in the original geocoded table will dynamically move the point associated with that record in the drawing. Adding a record to that geocoded table will create a new point in the drawing, and deleting a record will delete the corresponding point.

**Example**

Let us see how this all works in a specific example.

<table>
<thead>
<tr>
<th>Towns : Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>Longitude</td>
</tr>
<tr>
<td>119.47</td>
</tr>
<tr>
<td>-62.68</td>
</tr>
<tr>
<td>-123.42</td>
</tr>
<tr>
<td>-76.25</td>
</tr>
<tr>
<td>-77.05</td>
</tr>
<tr>
<td>-74.34</td>
</tr>
<tr>
<td>-70.84</td>
</tr>
<tr>
<td>-66.55</td>
</tr>
<tr>
<td>-104.79</td>
</tr>
<tr>
<td>-97.67</td>
</tr>
<tr>
<td>-122.11</td>
</tr>
</tbody>
</table>

Suppose we use **Access** to create an .mdb file with a geocoded table called **Towns** as seen above in a screenshot of an Access session. Each record has a latitude and longitude value as well as other information of interest for each town.

In Manifold, we choose **File - Link - Drawing** and then choose .mdb in the **Files of Type** box to open our Access .mdb file.
Manifold knows what we have is a geocoded table so it opens the Link MDB File dialog with appropriate options. Since we named our coordinate columns Latitude and Longitude Manifold can guess these are the ones we intend to use for our X / longitude and Y / latitude values so it will preload those column names into those boxes. We press OK.

The new components are created in the project pane.
If we pop open the linked table called **Towns** we see that Manifold simply linked a table into the project from the external Access .mdb table. There actually isn’t any data saved within the project file for this table: the data is brought dynamically into Manifold from the Access .mdb. If we changed something in the external Access .mdb, such as the name of a town, that change would also appear in the linked table as well.

![Linked Table Diagram](image)

This linked table in the project is what creates the linked drawing, using the **Towns Points** query as a dynamic translator.

If all we want to do is utilize the linked drawing we don’t need to understand how the query works or what it does. All we need know is that it transforms the simple geocoding style of data from the linked table into the more powerful geometry data form used by linked drawings.

However, for those who have a taste for spatial SQL and are curious about what that query does, we can pop it open in a query window and take a look.

```sql
OPTIONS CoordSys("Latitude / Longitude");
SELECT [Towns].*, AssignCoordSys(NewPoint([Longitude], [Latitude]), CoordSys("latitude / longitude")) [GeoData] FROM [TOWNS];
```

Setting aside for a moment the query telling itself it’s going to be doing things in **Latitude / Longitude** projection, we see the guts of the query use the **NewPoint** spatial SQL function to create a point from Latitude and Longitude coordinates. As is mentioned in the Spatial Extensions topic, the **NewPoint(x, y)** function "Returns a point with given XY coordinates in default coordinate system."

One very cool thing about linked drawings is that they can be created on-the-fly not just from tables, but also from queries. The query grabs the lat/lon data from the table and creates points from that data that are expressed as geometry data.

![Linked Drawing](image)
The linked drawing created from that query knows what to do with that geometry data. It makes points in the right positions, complete with all the other data that was in each record.

We can display the new Town Points Drawing in a map with our example US_Main drawing and see that the points are in the right place. Redmond is indeed up there in the rainy Northwest and Round Rock (the home of Dell Computer) is still in Texas.

Read Only Locations with Read / Write Data

One downside of linked drawings when they are created from geocoded tables is that they are basically read-only as regards position of the points. We can’t drag points around in the linked drawing and expect the latitude and longitude values to change in the originating Access .mdb file. If we change the latitude and longitude values in the originating geocoded table the points will move in the drawing, but we cannot move the points in the drawing and cause the values in the table to change.

However, and this is totally cool, linked drawings are read / write for all other fields in their originating geocoded table. If we pop open the linked drawing’s table in the Manifold project and change Round Rock to Red Rock, if we also pop open the Access .mdb file in Access, we will see that change in Access as well. If we change a value in the Access file we see the change in Manifold. Amazing!

Note: Changes will be displayed in Manifold whenever we trigger a refresh in an open window. This happens when a window opens or closes, pans or zooms or when we issue a View - Refresh command or, if we are using IMS if the web site has been told to refresh on some interval.

In fact, this is so much fun we recommend all Manifold users cobble up a geocoded table in Access, if only with three or four records in it, and then create a linked drawing from it in Manifold. Open Manifold in one window on your computer and open the linked drawing's table and also open the Access table in Access. Watch what happens as you change values in Manifold or in Access: the corresponding value will also change in the other window.
The screenshot above shows a Microsoft Access session opened with the *Towns* table from the *towns.mdb* file. Behind it is a Manifold session with a red arrow inserted into the illustration to indicate the point associated with the Round Rock, Texas, record. The Access table has that record highlighted.

If we delete that record in Access, we can see (after refreshing the view) that the point has disappeared from the Manifold drawing.

**Uses for Drawings Linked from Geocoded Data**

The intent of all this, of course, is not to have fun manipulating tables in one window while points move about or disappear from other windows. The great power we get from using drawings linked from geocoded tables is that we can keep data centralized within databases such as Access or SQL Server or even just tables in simple Excel spreadsheets and then use that data in GIS displays.

The beauty of this idea is that people or applications who know nothing whatsoever about GIS can continue to work with Access, Excel or our favorite DBMS program in their usual way, and automatically any changes they make to the data will be updated in our GIS display in Manifold.

Consider an example: Suppose we have a vehicle tracking application. Each truck in our fleet of trucks has a GPS device and a cell transmitter that reports the location of the truck. We would like to display the location of each truck in a web site that shows points at the location of each truck. When our truck equipment vendor sold us this system we also got some software that listens to the location reports and puts them into a table in our database.

To connect this table data to GIS and to the web we don't need to find a programmer who knows all about the truck hardware and all about databases and GIS and web programming as well.
Instead, we can divide the application into two parts using the common database table:

- **Acquire truck locations** - Just about everyone in the business of selling GPS/location devices for trucks provides software that will update an Access table or some other database table with a record for each truck and the current latitude / longitude location of that truck. That's easy to do and we probably have plenty of people on our staff who know how to do anything we want with the database software we've been using all these years. They might not know GIS, but they know their DBMS.

- **Display truck locations** - The Manifold part of the task is easy: create a linked drawing from the geocoded table that holds the truck locations and then do an IMS website that shows that drawing. The database and truck equipment people don't need to know anything about GIS, and our Manifold people don't need to learn how it is that the truck hardware communicates latitude / longitude locations into the database table. All they need know is the name of the table, the database it is in and the names of the columns that contain the latitude and longitude values.

The above approach is a great way of working with data from geocoded tables for many different types of applications. Whether we are keeping track of trucks, noting the locations of dealers in our retail network, grabbing the locations of recent earthquakes from a web site, marking the positions of favorite sushi restaurants or showing the sites of active fishing spots we can use the same method.

Some standard process with which we have long familiarity creates geocoded tables in our favorite database, which all our organization knows and loves. Without harassing anyone with GIS stuff, we use Manifold to create drawings linked from those geocoded tables. We can then work with those drawings in Manifold projects like any other, we can publish them in IMS web sites, we can serve them out as images using OGC WMS or a Manifold Image server, or we can use them to create things like PDFs from a print layout for publication. In all cases, whenever we use that linked drawing it will show the current data in the table at the moment of its use.

### Exchanging Files

If we would like to exchange data with other users we should employ linked drawings with care if the linked drawing is linked from a table outside the Manifold project. If the table controlling a linked drawing is outside of Manifold it will not stored within the Manifold .map project file. Sending a colleague a copy of our .map file will not also send them the database table from which the linked drawing is created.

### See Also

Linked Drawings
Create a Linked Drawing from a Geocoded Table
Formatting Lines in a Linked Drawing
About Geocoding
Geocoding
Street Address Geocoding
Spatial Geocoding with Match
Create a Map from a Geocoded Table
Project Pane - Paste as Drawing
Using Projections

Projections are used within GIS and mapping for three main reasons:

- To provide a more natural looking map,
- To enable measurements of areas and lengths in printed maps,
- To enable easy measurement in linear units such as meters or feet when performing analyses.

Manifold System has a very flexible approach to projections. In particular, the "native" projection in which the data of a particular component is kept can be different from the projection used to show that component within a map. Any projected component can be re-projected into a different projection at will.

Using projections does not require much thought or study if one uses projections in a simple way. However, it does require some reading of this documentation. After reading this topic, make sure to read the Projections topic and the Projecting a Map topic.

Projections and Maps

Maps can use any projection desired no matter what projections are used by the components within them. The view of the components within the map will be re-projected on the fly to suit the projection requested of the map. This works well so long as the components within the map are not too large.

To change the projection used by a map:

1. Open the map.
2. Launch the Edit - Assign Projection dialog.
3. Choose the projection to be used (see tips in the Projections topics).
4. Choose sensible parameters for that projection if parameters are required. Use the Suggest button to automatically enter reasonable parameters.
5. Press OK.

The above procedure changes only the view used by the map. It does not re-project the data in the components within the map.

Since dynamic re-projection requires processing time, as the components get larger it can take so long to re-project them on the fly that maps may become slow to display. A map can display faster if it uses the same projection as the components that make up the layers of the map. If one component is very large (such as an image) it may be helpful to use the same projection for the map as well as for that large component layer.

To use a layer's projection as the map's projection:

1. Open the map.
2. Right click on the layer tab for the component whose projection is to be used for the map.
3. Choose Use Projection.

The above procedure instructs the map to use the same projection as a particular component in the map.

Projections and Components

At times we may want to change the projection used by a component. For example, suppose we have a map that includes a compressed image layer that uses a very large linked image and it also has a drawing layer that uses a very large drawing. The map will use the same projection as the image because otherwise the linked image could not be displayed in the map (linked images cannot normally be re-projected). For speed, we might therefore want to change the native projection used by the drawing to be the same projection as used by the map.

Note: Before re-projecting a component into a new projection Manifold will ask us to verify that the starting projection was correctly assigned. See the Verifying Projections section below.
To re-project a layer to use the map's projection:

1. Open the map.
2. Right click on the layer tab for the component whose projection is to be used for the map.
3. Choose Project to Map.

At any time, we can change the projection of a component whether or not it is part of a map. For example, we may have a drawing using some UTM projection that we would like to re-project into Latitude / Longitude to allow export into some legacy drawing format. We can do so using the Edit - Change Projection dialog.

To re-project a component:

1. Open the component.
2. Launch the Edit - Change Projection dialog.
3. Choose the projection to be used (see tips in the Projections topics).
4. Choose sensible parameters for that projection if parameters are required. Use the Suggest button to automatically enter reasonable parameters.
5. Press OK.

The above procedure changes the constituent data within the component into the new coordinate system and units of measure used by the new projection. When re-projecting an image or a surface, the Suggest button will automatically suggest equal values for the X and Y local scale parameters, thus resulting in geographically "square" pixels.

Using the Suggest button will result in "reasonable" values for local scale, which will prevent the newbie error of commanding Manifold to re-project an image using pixel scales that result in a very large number of pixels being used to represent the image. Depending on exactly what the user specified, such an error could cause Manifold to re-project an image so that it requires many gigabytes of storage space, possibly exceeding the amount of free space available on disk. If you do not understand the meaning and proper use of scale and other parameters, it is easy to avoid such errors by using the Suggest button when re-projecting an image.

At times we would like to use a particular projection in a map and for faster performance would like to re-project the components that make up the layers of the map into that same projection. To do so, open the map and note the projection parameters used in the Edit - Assign Projection dialog. Next, open the component in its own window by double clicking on it and then use Edit - Change Projection to change the component's projection into the same projection and parameters used by the map.

**Very Important:** To re-project a component, use the Edit - Change Projection command. Do NOT use the Edit - Assign Projection command. This latter command is used to specify the projection to be used by a map window or to manually correct the projection of a component imported from a format that is unable to provide projection information.

**Verifying Projections**

Once a component has been correctly imported into Manifold, the system will automatically manage projections from then on. The catch is getting a drawing or image or other component correctly imported into Manifold in the first place if it must be imported from a format that does not reliably capture projection information. Surprisingly many formats for drawings and images used in GIS work are very old formats that do not reliably capture projection information.

It is so important to make sure that a projection has been correctly assigned that Manifold tries to ascertain when a component is imported from a format that does not store projection information. In such cases, Manifold will take special measures to ask us to verify the projection assigned. The first time such a new component is opened, Manifold will overlay an info bar onto the window telling us that the projection used by the component has not yet been verified, and inviting us to click on the info bar to verify the projection.

Clicking on the info bar will launch the Edit - Assign Projection dialog. We should take a moment to review the settings shown by that dialog to verify that the projection is indeed what it should be. If it is, we can click OK and thereafter Manifold will not nag us about verifying the projection. If the projection is not what it should be, we can specify the desired projection. Once we verify the projection assigned, we can open the component without Manifold nagging us to verify the projection.
We can get rid of the info bar without launching the Assign Projection dialog by clicking the X sign at the right side of the info bar; however, Manifold will know that we’ve not yet verified the projection of this component and the info bar will appear again the next time the component is opened in a window.

If we do not want to verify projections of new components, once we acquire enough expertise not to want to verify projections anymore unless it is absolutely necessary, we can get rid of the info bar by changing the Prompt to verify projections of new components option in the Tools - Options dialog.

Conversely, if we would like Manifold to remind us to verify the projection of all new components, even those imported from formats that correctly store projection information, we can uncheck the Suppress prompt for non-default projections option in the Tools - Options dialog.

**Note:** The rule of thumb used by Manifold to decide if a component has been imported from a format that does not store projection information is to simply raise the info bar whenever a new component's projection is Orthographic, the default used for imports from formats that do not provide coordinate information, or if the component has been imported in Latitude / Longitude with coordinate locations outside the expected range (+/- 90 latitude and +/- 180 longitude).

That leads to some unnecessary use of the info bar as occasionally we will import a component from a perfectly expert GIS format where the component just happens to have truly been stored in Orthographic projection, but for a short hand rule of thumb it works surprisingly well.

More worrisome is that the rule of thumb will miss occasions when formats have some projection information but not complete projection information. The classic case is shapefiles using incomplete .prj accessory files or some image formats accompanied by “world” files. In that case, there may be enough information to cause a projection other than Orthographic to be used, so that the rule of thumb does not cause the info bar to be raised, but not sufficient information from the format to have all projection parameters correctly assigned. In that case, we have be alert enough as users to manually launch the Assign Projection dialog to verify the projection and to accurately specify all parameters if needed.

Beginning GIS users working with formats such as shapefiles or graphics formats using "world" files may want to uncheck the Suppress prompt for non-default projections option in the Tools - Options dialog so that they are reminded by Manifold to always verify projections on all new components.

**XML Files Created upon Export and Used on Import**

Not all formats to which Manifold can export have the ability to correctly save projection information. As a safety measure to ensure that projection information is never lost, Manifold always writes an accompanying .xml file when exporting drawings, images, labels, maps or surface components to a file. The accompanying .xml file contains coordinate system information and some other metadata. The XML file is created for all formats, even those that correctly save coordinate system information.

When Manifold imports drawings, images, labels or surfaces, the system will check for an accompanying .xml file that might have been written by Manifold. If such an .xml file exists, Manifold will read it and use the information it contains to load a correct coordinate system.

In addition to the .xml file that is automatically created, Manifold projections dialogs such as the Edit - Change Projection dialog provide toolbar commands to manually write or read projection information from accessory files using generic XML, ERDAS AUX, Golden Software GSR or ESRI PRJ files.

**Automatic use of Custom Datum Transformations**

Manifold uses high accuracy coordinate transformation mathematics when re-projecting data from one coordinate system to another. In certain parts of the world, custom transformation formulae are routinely used (and, at times mandated by law for certain uses) to convert datums during re-projection. If the Use custom datum transformations option is checked in the Tools - Options - Miscellaneous pane (checked by default) Manifold will use such custom datum transformations when available.

Currently, NADCON formulae are used to convert between NAD27 and NAD83 in North America and NTv2 formulae are used to convert Canadian, Australian and New Zealand datums supported by NTv2.

If a converted location is outside of the conversion domain supported by such custom methods, no datum conversion is done.
**Datum Pairs supported by Custom Transformations**

<table>
<thead>
<tr>
<th>Pair</th>
<th>Transformation</th>
</tr>
</thead>
<tbody>
<tr>
<td>North American 1927 (mean for CONUS)</td>
<td>&lt; NADCON &gt;</td>
</tr>
<tr>
<td>North American 1983 (mean for CONUS)</td>
<td></td>
</tr>
<tr>
<td>North American 1927 (Alaska excluding Aleutian Islands)</td>
<td>&lt; NADCON &gt;</td>
</tr>
<tr>
<td>North American 1983 (mean for CONUS)</td>
<td></td>
</tr>
<tr>
<td>Old Hawaiian (Hawaii)</td>
<td>&lt; NADCON &gt;</td>
</tr>
<tr>
<td>North American 1983 (mean for CONUS)</td>
<td></td>
</tr>
<tr>
<td>North American 1927 (mean for Canada)</td>
<td>&lt; NTv2 &gt;</td>
</tr>
<tr>
<td>North American 1983 (mean for CONUS)</td>
<td></td>
</tr>
<tr>
<td>Australian Geodetic 1966 (Australia, Tasmania)</td>
<td>&lt; NTv2 &gt;</td>
</tr>
<tr>
<td>Australian Geocentric 1994 (GDA94)</td>
<td></td>
</tr>
<tr>
<td>New Zealand Geodetic 1949 (New Zealand)</td>
<td>&lt; NTv2 &gt;</td>
</tr>
<tr>
<td>New Zealand Geodetic 2000</td>
<td></td>
</tr>
</tbody>
</table>

Custom datum transformations will be invoked **only** when converting between the above pairs in either direction. For example, they will be used when converting NAD27 to NAD83, or when converting from NAD83 to NAD27. Pairs shown above are listed using their full names as they appear in the datum box in Manifold projection dialogs, not using the abbreviated form commonly used. For example, **North American 1983 (mean for CONUS)** is referred to as **NAD83** in abbreviated form.

The **names** of the above datums are significant, not the parameters used. For example, if we convert **GRS80** (the parameters of which are identical to NAD83 so it is the same datum) to NAD27 (mean for Canada), Manifold will **not** use the custom NADCON conversion formulae. This is done on purpose, so we can use NAD83 as the name of the datum for conversion-sensitive North American data (so that the custom transformation will be used) and GRS80 as the name of the datum for all other data where we would like to use the general purpose routines.

**Custom Datum Pairs for NTv2**

Additional datum pairs may be added to those supported by Manifold for use in NTv2 transformations through customization. See the Custom Datum Grids for NTv2 topic for details.

**A Note on Orthographic Projection**

Manifold uses the Orthographic projection by default for components imported from formats that do not provide any projection information. Although the Orthographic projection is the classic “view from space” projection and thus widely used, one limitation of Orthographic is that the projection is defined only for the sphere datum. If other datums are to be used, an alternative “view from space” projection is the Stereographic projection.

**See Also**

- Projections and Imported Components
- Projections
Projections and Imported Components

Manifold can import drawings, images, terrains and tables from a very wide range of formats. Some formats such as SDTS are very modern formats that are easy to import with little user intervention. Other formats are very primitive formats that are poorly suited for data interchange. Some formats well suited for CAD programs are less well suited for GIS. Primitive formats or those formats not originally designed for GIS can require a lot of user intervention to import into a GIS program like Manifold.

When working with modern formats the import process is normally easy and automatic in all respects. For example, Manifold can easily import drawings from MapInfo mid / mif format in almost all cases without any need for user intervention. In such cases we import the drawing, pop it open, verify the projection by launching Assign Projection and clicking OK and we're done.

When working with archaic GIS formats such as ESRI.shp or with formats optimized for CAD work such as AutoCAD.dxf user intervention often will be required. In that case, we may not get away with just verifying the projection by pushing OK in the Assign Projection dialog. Instead, we will have to actually understand what the projection parameters are supposed to be and to change them to the right parameters if necessary. That requires a lot more thought than simply pressing OK.

The most frequent class of technical support questions when using Manifold arise from confusion over importing projected maps from old GIS formats such as ESRI.shp or non-GIS formats such as AutoCAD.dxf. Because such formats require a lot of user involvement when used with projected maps, they require a much greater degree of user understanding for correct import. That's a drag, because it requires the novice user to learn much more about projections than seems fair for a beginner. On the plus side, we don't have to learn very much in order to take advantage of the very large amount of free GIS data available in legacy formats.

This topic is an overview for reasonably experienced users. If you are completely unfamiliar with projections and coordinates you should also read the Projections book topics, beginning with the Projections topic. If importing shapefiles, see also the Import a Shapefile example.

Projections and Older Formats

Problems with old formats usually arise if the data they contain is intended to represent a projected map. If the data represents an unprojected map we can breathe a sigh of relief. Even very old formats can usually be imported quickly if the data they contain is not projected data. However, when old formats are used to save projected maps we will almost always have to participate in the import process by specifying projection parameters manually using the Assign Projection dialog. This is especially true of non-GIS formats such as .dxf that were designed to save CAD drawings. Regardless of how convenient they are for CAD usage, such formats will require extra steps to import into GIS.

Modern GIS formats are "smart" and automatically save the projection parameters in use together with the data. During import, Manifold will fetch all necessary parameters from such modern formats automatically and will load the coordinates properties for that drawing with the correct parameters necessary to use the data.

CAD formats such as AutoCAD.dxf and older GIS formats such as ESRI.shp, do not save the projection parameters in use. While one is tempted to refer to such formats as "dumb" formats in contrast to the "smart" encoding of projection information within modern GIS formats, we suggest the term "legacy" format. After all, such formats work perfectly well when used as originally designed for their intended purpose (usually a more limited purpose than expected of modern GIS applications) and should not be blamed if someone uses them inappropriately.

Importing data from legacy GIS formats causes so much trouble for beginners in GIS that Manifold tries to protect us from newbie errors by reminding users to verify the projection in use when a new component is first opened if Manifold thinks there is a good chance the component has been imported from a format that does not save projection information. Manifold raises an info bar to ask us to verify the projection if the component has been imported in Orthographic projection or if it has been imported in Latitude / Longitude with coordinate locations outside the expected range (+/- 90 latitude and +/- 180 longitude).

The classic error is that someone imports projected data from a legacy format like shapefiles and then when they see it looks OK visually they assume it imported OK as well when in fact the correct projection still needs to be assigned. When an info bar pops up reminding us to verify the projection it is more difficult to skip over that necessary step.

When legacy formats are used to save projected maps they require users to keep track manually of the projections used in the file. This is usually done with a readme.txt file, a "metadata" information text file or other
documentation that accompanies the data files and specifies what projection parameters should be used with those files. Users must read the documentation to know what parameters to use. If we download the data files and neglect to fetch the documentation that says how to use them we will find ourselves in trouble. Of course, we will also be in trouble if the authors of the data files neglected to provide any accompanying documentation for us to read.

Because the above is such an obvious hassle most sensible people will refrain from using CAD formats or older GIS formats to save projected data. GIS data in such formats therefore is usually unprojected data that may be imported into Manifold with no special user action required except to specify which datum should be used (as shown in the Import a Shapefile example).

On those occasions when we encounter a projected map that is saved in an older format we will have to specify projection parameters after the import. We can do this using the Assign Projection dialog when we verify the projection to be used.

However, there may be no warning that the data contained in the file is projected and so must be interpreted in a way known only to its author. When acquiring data in older formats we should always keep in mind the possibility that the files contain projected data. If we suspect projected data is involved, we should keep an eye out for documentation that explains what projection parameters are to be used with the file. We may have to search long and hard for this information.

To import from a format that does not save projection information:

1. Import the drawing or image using File - Import. Use default settings.
2. Open the drawing or image.
3. Click verification info bar to use the Edit - Assign Projection dialog to specify the projection information that should be used.

Note that complete information on all projection parameters is required. If a projected drawing is supposed to use Lambert Conformal Conic projection then simply saying that it uses Lambert Conformal Conic projection is not enough: all of the various parameters used within that Lambert Conformal Conic projection (such as datum, latitude / longitude center of the projection, standard and any optional parallels used) must be known as well and specified in the Assign Projection dialog.

Users should be careful to get the best information they can on the projections supposedly used by a given drawing or image. Be skeptical and be thorough: publishers of projected data using legacy formats may not realize that the information they provide is inaccurate or insufficient. For example, if one imports projected files in .shp format that show Wisconsin, it is not enough to say the files are in "Wisconsin State Coordinate System (NAD27)." If the files are in the State Plane Coordinate System using NAD27 we need to know whether they are in the Wisconsin North, Central or South zone.

Another issue is that a projected drawing imported from a legacy format may appear to have imported well at first glance. There is not much visual difference between projections and "unprojected" Latitude / Longitude coordinates in many parts of the Earth when small regions are in view. If we don't use the imported drawing together with other drawings it could take us a while to realize that someone fooled us by providing a projected data set in a legacy format.

It is also quite possible that a Latitude / Longitude unprojected drawing imported from .shp will look absolutely perfect and even will align well with other drawings. However, the datum used might be different than the default datum of WGS84 used in most modern GIS settings, such as GPS devices and Manifold. For example, many drawings in .shp format published by USGS will use the NAD27 datum. If we forget to change the datum after import we could introduce a small but possibly significant error into the data.

The message from the above is that when we verify the projection used by a new component we should not just launch the Assign Projection dialog and then blindly, credulously click the OK button without taking a moment to carefully review the dialog.

**Manually Specifying Projection Information**

Assuming we can lay our hands on the necessary projection information, we can import projected drawings or images from legacy formats. We first import using default settings for that format and then change the projection parameters from their default settings into the correct projection parameters to be used with that data. This is done on a one-time basis using the Edit - Assign Projection dialog.
To do this, click on the new drawing to highlight it in the project pane, and then open the Edit - Assign Projection dialog (or, open the drawing and then click on the verification info bar to open the dialog). This will display the (wrong) parameters used during default import. Enter the correct parameters intended to be used with that data. We only need do this once and then forever more Manifold will keep track of all projection parameters for us automatically.

Once a drawing or image is imported, the projection parameters in use for that component may be seen at any time by opening the Edit - Assign Projection dialog for that component. We can also reach this dialog by viewing the component’s coordinate properties using View - Properties.

Changing the projection assigned to a drawing simply changes how Manifold interprets the internal coordinate numbers that comprise the actual data set. The only time we would do this manually is if we import a projected map from a legacy format and need to tell Manifold (manually) what parameters are supposed to be used. When importing a projected map from a legacy format, the internal coordinate numbers are already correct but there is nothing in the format that says how these numbers should be interpreted. We can tell Manifold how those numbers should be interpreted by entering them into the Assign Projection dialog on a one-time basis.

**Importing Images**

Regarding projections, most images fall into one of several classes:

- Simple overhead photographs created without any projection intended.
- Unprojected images intended for geographic usage.
- Projected images saved in smart geographic image formats.
- Projected images saved in legacy formats.
- Linked images served by an image server or linked from a data source.

Most images one encounters are simple photographic images in .jpg or other photo image formats. They usually show overhead aerial photographs of small geographic areas and are frequently oriented “north up.” They are not intended as “projected” data sets, although most such images can often be handled as if they are projected in the Orthographic projection.

The Orthographic projection shows geographic data as a "view from space.” When a map of the world or a hemisphere is shown in this projection, one sees the Earth as it would be seen from space. Zooming far into the center of such a projection provides a simulated ride towards the planet from space to a location directly overhead a particular location.

For all practical purposes, an aerial photograph shot from directly overhead a particular spot that covers only a few kilometers or miles may be treated as an image in Orthographic projection. By default, unless something about the format used tells Manifold to assign a specific projection, Manifold imports all images as if they are in Orthographic projection and automatically georeferences their lower left corner to the intersection of the Equator and the Prime Meridian (that is, zero latitude and zero longitude).

**Note:** Although the Orthographic projection is the classic "view from space" projection and thus widely used, one limitation of Orthographic is that the projection is defined only for the sphere datum. If other datums are to be used, an alternative "view from space" projection is the Stereographic projection.

To use North-up overhead images as part of a geographic map, we could use the Edit - Assign Projection dialog to "move" them to the right location. We could do this by moving the center of their projection (the lower
left corner, in the case of images) as necessary to the right location and by adjusting the scale of the image. An easier way to accomplish the same thing is to use Manifold’s georegistration tools to easily move, scale and warp the image as necessary from its default location. This is a lot easier than editing the coordinates properties by hand using the **Assign Projection** dialog.

Some images intended for geographic use cover very large areas of the Earth and are presented in "unprojected" **Latitude / Longitude** form. Using such images in Manifold is easy: we import them using default settings for images and then change the projection assigned in the **Edit - Assign Projection** dialog for that image to **Latitude / Longitude**. We will also have to set the scale factor in use. The scale factor can usually be determined from documentation or by computation (divide the number of degrees covered by the image by the number of pixels in the image to see how big each pixel is supposed to be). If desired, we can use the georegistration tools in Manifold or simply use trial and error to adjust the scale factor to be a good match to a known vector map.

Projected images that are saved in smart geographic formats are also easy to use. Manifold will import them with all necessary parameters with no user intervention required. All we need do is verify the projection used and we’re done.

Projected images that are saved in legacy formats will waste more time than all the above cases put together. Such images usually fall into two subclasses:

- Images that are close to, but are not quite, overhead views and so must be adjusted.
- Sophisticated projected images saved in legacy formats that do not reliably capture projection information.

The first case is easy to deal with. We ignore the published projection information (if any) and assume the image is in Orthographic projection and then we force it into a shape that matches a known vector map by using Manifold tools to warp the image. When done in a sophisticated way, this is known as **orthorectification**. Many Landsat images, for example are very close to perfect overhead views. Even though such images are technically in Space Oblique Mercator projection, it is often much easier to treat them as Orthographic projection and to warp them slightly using Manifold georegistration tools than it is to track down the exact projection parameters that must be used.

The second case is more difficult to deal with and is directly analogous to encountering a projected drawing saved in a legacy GIS format. We need to find the projection parameters used in the projected image. We can then enter them into the **Edit - Assign Projection** dialog for this image. The problem here is that few GIS packages can handle projected images so often webmasters or librarians or other people who are involved in distributing such images realize that that projected images should be accompanied by projection parameters. Most folks will just treat a Landsat image, for example, as a simple photograph. When we ask for a Landsat image of a given area they simply give us a **.jpg** and don’t realize that without the accompanying projection information it is not very useful in a GIS.

We can open such images in Manifold, but if we don’t have the projection information we will not be able to re-project the image or otherwise use it in a sophisticated way. In the case of Landsat images, if we can track down the original copy of the image in the Landsat database we have a good chance of finding the projection parameters that must be used to use this image as a projected image within Space Oblique Projection.

**Changing Projections**
Once a drawing or image is correctly imported into Manifold and the projection has been verified we can change the projection into whatever projection we like. To re-project a drawing into a new coordinate system we use the Edit - Change Projection dialog.

The dialog opens with the projection parameters currently in use. These are taken from the coordinates properties of the drawing. If the drawing was imported correctly (with manual updating of coordinate properties if need be using the Assign Projection dialog), these parameters will be the correct parameters required for the coordinates in the drawing to make sense.

When we use the Edit - Change Projection dialog to cast the drawing into a new projection, Manifold will re-compute the coordinates within the drawing to the equivalent numbers required by the new projection. Manifold will also change the coordinates properties to the new parameters now required to make sense of the new numbers. All of this happens automatically. Henceforth, when we open either the coordinates properties dialog or the Assign Projections dialog we will see new parameters to match the new coordinate numbers. Re-projecting a drawing in this way permanently changes the internal coordinates used to define the drawing.

Because drawings are often seen through a map window within maps, it is not always necessary to re-project them if we want to see the drawing in a different projection. We can leave the drawing in whatever projection it was in on import and view it using a map that has been assigned a different projection. The map will re-compute the view on the fly to present the drawing as it would appear in that different projection.

Specifying the projection used by a map does not change any data inside the drawings - it simply changes the way the data is seen by computing on the fly how the drawing should look in the projection requested of that map. Even if several different drawings using different native projections participate in a map Manifold can re-compute them all on the fly to display them correctly in the projection desired for that map. As long as the drawings are not so large that our computer slows down too much during the re-computation process, this works well.

If we are working with large drawings and they must appear within a map using a different projection the process of displaying them in the map may be too slow for our taste. In that case, we can either change the projection of the map to be the same as the drawing, or we can re-project the drawing into the projection used by the map. A map window can work faster if the drawings and images they contain use the same projection requested of the map. This is especially true if many large drawings and images are used in the map.

As explained in detail in the Images topic, images may also be re-projected Images that are used within maps will often be re-projected to match the projection requested of the map. This is because images tend to involve much more data than drawings (each pixel in an image is like an object in a drawing) and re-projecting them on the fly is slower than re-projecting drawings.

**A Projections Strategy**

For most interactive editing and map preparation work in middle latitudes a good strategy is to keep all drawings in Latitude / Longitude form and to work with them in maps using the Latitude / Longitude projection. This is the name given to the "unprojection" that uses simple latitude and longitude degrees instead of any other projection. Most GIS data is published in this form, so it's likely that this will be the native "projection" used by most drawings on import. Latitude / Longitude projection is also used to publish most images that cover the entire Earth, such as the globe.bmp sample image shown above.

When working within smaller regions or individual cities, working in Orthographic projection centered on the region of interest is often a good choice. This is the default projection used for image and non-geographic drawing import, so it is fast even if images are involved. It is also the closest match to an overhead view as would be seen from an airplane flying over the region.

Of course, if you have a fast system or are working with smaller maps you can simply keep drawings and images in whatever projection they were in when imported and use whatever projection you like in maps.

**Very Important:** To re-project a component, use the Edit - Change Projection command. Do NOT use the Edit - Assign Projection dialog. This latter command is used to manually correct the projection of a component imported from a format that is unable to provide projection information.

**Notes**
Introduction

As discussed in the Coordinates topics, CAD drawings coming into a map from non-geographic contexts will have to be georeferenced the way images must be georeferenced. Like images, CAD drawings also use the Orthographic projection by default and are georeferenced in the same way.

Recent extensions to ESRI .shp format will save partial projection information within a .prj auxiliary file; however, most .shp data predates the availability of this latest hack so it does not help much.

Although the georeferencing tools are a lot easier to use than editing an image's coordinates properties by hand, every expert user of Manifold should experiment with hand-editing coordinates properties to better learn how coordinates work.

We must apologize for such a large, dense topic so early in the Introduction. However, it is made necessary by the prevalence of legacy GIS formats. Working in smart GIS formats is easy and requires much less expertise in projections.

See Also

Projections and Legacy Formats
Edit - Change Projection
Edit - Assign Projection
Import a Projected Shapefile
How to Print

Manifold provides two paths to printing:

- The contents of any component window may be printed in a default way with File - Print. The visible contents of the window will be printed using a simple dialog.
- More sophisticated layouts may be composed using a Print Layout which may then be opened and printed using File - Print. The layout will be based on whatever settings (paper size, orientation, etc.) are specified in the File - Page Setup dialog.

To Print using a Print Layout

1. Click in the project pane and create a new layout with File - Create - Layout. The Create Layout dialog will pop open. If the layout is to be based on a particular component (often the case when we create a layout for a single component), choose the component to be used in the Create Layout dialog.
2. Open the new layout.
3. Verify paper size and other overall options are correctly specified in File - Page Setup.
4. From the project pane, drag and drop into the layout any other components that are to appear as elements of the layout. Create any new layout elements, such as text entries, using layout tools.
5. Edit the elements in the layout to change size, position, views to be printed, etc. Clicking on an element selects it so that it may be formatted using the formatting toolbar. Right clicking on an element allows choosing the Properties dialog to change the views and other display characteristics. CTRL-ALT clicking on an element will select it for editing so that it may be moved or resized.
6. When satisfied with the appearance of the layout, use File - Print to print the layout.

Note that printing items with a layout is a three-step process: create the layout, adjust the elements of the layout to taste, and then print the layout.

Example

In this example we create a new layout from a drawing.

We begin with a drawing of Mexico. We would like to print this drawing using a layout.
In the project pane toolbar we click on the Create button down arrow for a pull-down menu of project pane components that may be created.

We choose a Layout. This will pop open the Create Layout dialog.

The Create Layout dialog displays a list of all components in the project. This project only has one component, the Mexico drawing. We click on Mexico Drawing to highlight it and then press OK. If we wanted our new layout to be named something other than "Layout" we could have specified a new name as well.

In the project pane a new layout is created underneath the parent Mexico Drawing component. To open the layout, we double-click on it.
The default layout will show the entire component on whatever is the default page size and orientation defined for our printer. Since most people use Letter or A4 sized paper in a portrait orientation it is quite likely the layout will look like that above. The light gray lines bordering the drawing show the extent of page margins that have been set for this printer.

Given the East - West extent of Mexico, our drawing would look better if printed on the paper in landscape orientation.

To change the paper orientation we choose File - Page Setup and click on Landscape, then we press OK.
The layout will be redisplayed using landscape orientation for the paper. If we like, we can print the layout now with File - Print. However, to continue this example we will add a text element to the layout.

We can insert text into the layout using the Insert Text button.

Click on the Insert Text button and then draw a box in the layout where the text is to appear.
In the **Insert Text** dialog enter the text desired and press **OK**. In this case, we will simply enter the word "Mexico". If we later change our mind about the text we can always edit it by double-clicking the text element. This will reopen the **Insert Text** dialog for editing. See the **Layouts** topic for information on other parts of the **Insert Text** dialog.

The new text appears in the layout using the default Tahoma 9pt font, which is too small to be legible in the illustration of the layout shown above. In any case, we would like to make the font larger. To do so we will select the new text element and change the font size using the formatting toolbar.

Before changing the font size we will first change the selection style used. Manifold can show selections in different styles, with the **Dense Dots** style being the default. It may be more convenient to show selections in layouts using the **Border** style.

To change the selection style we click on the down arrow of the **Selection Style** button in the selection toolbar.
We then choose **Border**.

To select the text element that was entered we click on it.

It immediately appears within a red selection box that shows the extent of the text element box we drew with the mouse.
To change the font size, we click on the font size button in the formatting toolbar and choose a new size. We can change the font to a 40 point size, the font to **Verdana** and push in the **Bold** button to boldface the font.

The result is a text element exactly the way we want it. To turn off selection we can simply press the **Select None** button or just click in the layout window outside of any layout element.
To print the layout we choose **File - Print**.

### Exporting to a File

To save the layout to a file, we choose **File - Export - Image**.

![Export dialog](image)

In the **Export** dialog we choose the file name to which to save and the file type. In this example we will save to a **.pdf** file.

![Export PDF File dialog](image)

In the **Export PDF File** dialog that follows choose the resolution to use for the output file. **Pixel resolution** specifies the resolution with which images and surfaces that are in the layout will be saved. **Vector resolution** specifies resolution used to save drawings and labels as well as layout elements such as text or horizontal and vertical lines. If saving to PDF for subsequent work in some other program before printing choose a **Vector resolution** setting that is the same as the resolution of the printer that will be used. If printing to a 600 DPI printer, for example, use 600 DPI. **Pixel** and **Vector resolution** must be at least 1 DPI and cannot exceed 7200 DPI.

The **Ignore styles** checkbox is an advanced option that forces Manifold to export the image using default point, line, area and label styles. This can be useful when exporting for later use in vector-style graphics editors such
as Illustrator because the default styles produce a relatively small number of simple vector objects per Manifold entity. In contrast, more complex styles available in Manifold can produce five to ten relatively complex vector objects per point. If the Illustrator operator prefers to work with simple objects check this box.

See the Export Layout - PDF topic for details on other options in the Export PDF File dialog.

The .pdf file can then be used in other programs. For example, it can be viewed in Adobe Acrobat Reader as shown above or imported into Adobe Photoshop.

Note: Although .pdf and .ps formats are often thought of as "universal" formats that can capture exact renderings of data for use in other programs, they do have numerous limitations. For example, neither .pdf nor .ps can render opacity. If a layout includes a map that shows a drawing below an image that has 50% opacity in the map, the .pdf created by exporting the layout to a .pdf file will show the image with 100% opacity. .emf exports do support opacity for images and surfaces but not for drawings.

Moving Elements

Elements can be moved by selecting them for editing with a CTRL-ALT-click and then dragging them into the desired position.
To move the text element, hold down the **CTRL** and **ALT** keys while clicking on it. Edit handles will appear on the element. The mouse will automatically change to a move or resize cursor when positioned over the text element or over an edit handle.

To move the element click on it and drag to a new position.
Note that text elements have transparent background color by default, as seen in the background color box in the formatting toolbar.

The transparent background color allows text to be positioned over other layout elements with the text "floating" above the other elements. If desired, the background color for the text element can be changed to some other color, to fill in the rectangular area of the text element with background color.

**Visual Perception**

Printed layouts may look slightly different (especially if varying opacities are used in maps) in terms of brightness, opacity or contrast than they appear on screen. This effect arises from the wildly different nature of the media used, a computer monitor versus a printed page. After all, images created by emitted light, like those shown on a computer monitor, are very different from those created by absorbed or reflected light, like those shown on a printed page.

For that matter, even using different types of paper will change the appearance of a printed page at times. Using photo quality paper in inkjet printers can result in dramatically brighter and more vivid images that have better contrast than images obtained with non-photo quality paper.

**Notes**

Why bother changing the selection style to **Border**?

Some users feel that the **Border** selection style is less visually intrusive when formatting text and other elements in layouts. At left above is the **Border** selection style while at right above is the default **Dense Dots** selection style. Either way is fine according to the taste of the user.

**Tech Tip**
In the Tools - Options dialog check the Adjust display scale for monitor resolution box. That will match the scale used in Manifold to what is literally shown on the monitor given monitor size, DPI and other factors.

When working with layouts we can then choose View - Zoom to and choose a zoom of 1:1 so that the layout is shown exactly real size. If we have a large enough monitor we can then see the layout exactly as it will be printed with, for example, an A4 sized paper layout being displayed as a full-sized A4 sheet of paper on the monitor.

See Also

Layouts
Exporting Layouts
Export Layout - PDF
Printing to Files or Images

Maps created in Manifold will often include combinations of drawings, images, labels, surfaces and other components. For export to other programs (such as PhotoShop or Illustrator) we may wish to save a given map as an image with higher resolution than that visible on screen.

To do so, begin by creating a layout for the component. There are then two choices for creating an image from a layout:

- Use the Tools - Make Image command to save the layout as an image in the project. When called from a layout, Make Image allows specification of the desired size of the image in pixels. This command can be used to save a drawing or other component as an image (a process often called “rasterization” in some software) or it can be used to save a complete map as an image.
- Use the File - Export - Image command to save the layout to an image file. Use this command to save a layout as a .pdf or other file format.

To save a map or other component as an image in the project:

1. Create a print layout for the component.
2. Open the layout and choose Tools - Make Image.
3. In the Make Image dialog, choose a name for the image, a description if desired and the resolution desired. Press OK.

Note: We can also right click onto a component in the project pane and choose Make Image or Export. Use Export - Image to export to Adobe .ai (Illustrator), .ps (PostScript, as used by PhotoShop or Illustrator) or .pdf (Acrobat reader) files.

Make Image Options

- Name: Name of the image to be created.
- Description: Optional description to be entered into the image component's Description property.
- Resolution: Desired resolution in DPI. Appears only when making an image from a layout.
- Size: A status report at the bottom of the dialog gives the size of the image to be created in megabytes given current DPI settings.

To save a map or other component as an image file:

1. Create a print layout for the component.
2. Open the layout and choose File - Export - Image.
3. In the Export dialog, choose a name for the saved file and the type of file to be created. Press Save.
4. Some file types (such as .ai, .pdf or .ps) will raise dialogs to allow choice of Pixel resolution or Vector resolution and other options. Choose the options desired and press OK.

About Pixel and Vector resolution

Layouts can consist of many different types of elements. They can contain components such as images, surfaces, drawings, labels and other components, system-created items such as North arrows and legends, and elements created in the layout dialog such as text and horizontal and vertical lines. When a layout is exported to an image format all of these different types of elements must be converted into a common form that will be understood by the destination file format.

In the case of image formats, even quasi-vector formats such as .pdf, the various different elements must be rasterized into some specific level of detail or resolution. The Pixel and Vector resolution settings control how
this is done. **Pixel resolution** is used to convert for export any images and surfaces that are in the layout. **Vector resolution** is used to save drawings and labels as well as layout elements such as text or horizontal and vertical lines.

There is no resolution gain in using a higher pixel resolution than the native resolution of any images. If an image was scanned at 150 DPI there is no point in exporting it at 600 DPI since the higher resolution will simply use 16 small, 600 DPI dots to duplicate a single, larger 150 DPI dot. It may be convenient to save an image at higher resolution simply to accommodate later work in a program like PhotoShop.

If saving to a file for subsequent work in some other program before printing choose a vector resolution setting that is the same as the resolution of the printer that will be used. If printing to a 600 DPI printer, for example, use 600 DPI.

**See Also**

**How to Print** for an example of saving a layout to a file.

**Layouts**

- Export Layout - AI
- Export Layout - EMF
- Export Layout - PDF
- Export Layout - PS
Working with GPS Receivers

The Global Positioning System (GPS) provides a low cost and high accuracy means of finding the exact location and altitude of any place on Earth using inexpensive, hand-held GPS receivers. GPS receivers costing well under $200 can provide 15-meter accuracy and GPS receivers enhanced with WAAS capability that cost slightly more can provide better than 3-meter accuracy. Connecting a GPS receiver to a laptop running Manifold provides a high quality means of automatically acquiring position and other GPS data at a very low cost.

Manifold’s GPS Console provides direct interaction with a GPS (Global Positioning System) device connected to the computer via a serial port or USB connection. The GPS Console allows setting moving map mode or automatically fetching position and other data from the GPS device.

The Manifold GPS console works with any GPS receiver or other device that can communicate with a PC using non-proprietary, standard NMEA version 2 protocol using the sentences enumerated at the end of this topic, or with Garmin devices using either Garmin protocol or NMEA protocol. See the notes below under the heading Choosing a GPS Device for more information.

To acquire data from a GPS receiver:

1. Attach an NMEA compatible or Garmin protocol GPS receiver to a serial port or USB port on the computer. If a serial port connection is used, note which serial port (COM1, COM2, etc) is used. If you are using an NMEA compatible device, make sure it is set to NMEA version 2.
2. Launch Manifold. Open the project into which the GPS Console will save objects. Create a drawing if one does not yet exist. We must have at least one drawing in the project to accept data from the GPS device to use Track commands.
3. Open the GPS pane with a SHIFT-ALT-G or with a View - Panes - GPS Console command from the main menu.
4. In the GPS Console, choose the Drawing into which objects will be saved. The drawing cannot be a read-only drawing (such as a read-only drawing on an Enterprise server in Enterprise Edition).
5. Press Properties in the GPS Console toolbar. If a serial port is used, choose the serial port and the serial port properties for the GPS device. Most GPS devices use the default baud rate of 4800 and other default settings. If a USB port is used, choose either the Garmin USB connection or the generic USB Human Interface connection if a non-Garmin USB GPS is in use. Press OK.
6. Press in the Connect / Disconnect button to connect to the GPS device.
7. Press in the Track option buttons for the GPS tracking actions desired, typically the Track Points or Track Line button as well as the Track Columns button. If Track Columns is used, choose the destination fields within the drawing’s table for the various data fields acquired from the GPS receiver.
8. Press Track to manually acquire one reading of data from the GPS receiver as set forth by the Track options.
9. To automatically acquire repeated readings of data, choose the Refresh interval and press the Track Sequence button.
10. Manifold will acquire readings of data as set forth by the Track options until the Track Sequence button is pushed out again.

Note that the GPS Console can write to a drawing even if it is not opened. Once GPS tracking is started by pushing in Track Sequence, tracking will continue until the Track Sequence button is pushed out even if the GPS Console pane is hidden.

The Track action buttons work independently of each other, except that Track Columns requires Track Points to be enabled. If the Track Points button is engaged the system will create points whether or not the Track Center or Track Line buttons are engaged. Push in the Track Line button and the system will begin growing a line to follow changes in position as well as creating points. Push out the Track Line button and the system will stop growing the line but will continue to create points. Push the Track Center button in or out at any time to start or stop "moving map" mode where the drawing is centered to the current GPS position.

Most people working with the GPS Console will work with a map that shows some sort of base map drawing in one layer with a blank drawing into which GPS points or lines are written as a second layer. The base map layer provides orientation for the user while the new drawing layer keeps the data created by the GPS conveniently organized in a second layer.

Controls
Connect / Disconnect - Press IN to connect to the GPS device using serial port parameters specified in the GPS properties dialog. The GPS dialog cannot fetch data from the GPS device until it is connected. Connecting to the GPS also serves as a test of the connection.

Track Center - “Moving map” mode: with each reading of data from the GPS receiver move the drawing so that the acquired position of the GPS is always in the center of the currently active window. When Track Center is engaged and either a Track or Track Sequence command is issued the active window will be panned to center on the location reported by the GPS.

Track Line - With each reading of data from the GPS receiver extend a line to the current position. This option creates a line object that shows the sequence of tracked positions.

Track Points - With each reading of data from the GPS receiver create a point at the current position. This option creates a series of points that show the sequence of tracked positions.

Track Columns - Used to choose which names will be used for destination fields of downloaded data. Otherwise, has no effect unless Track Points is selected. Raises a dialog allowing choice of the destination fields within the drawing’s table for the various data fields acquired from the GPS receiver. Choose a blank destination in the Save To column to not import a given GPS field. When selected, Track Columns will write the selected data fields into the specified destination fields in the drawing’s table with each reading of data from the GPS receiver.

Track - Press to fetch data from the GPS. Each press of this button commands one acquisition of data and the creation of objects or moving map action as specified by the Track Center/Line/Points/Fields buttons.

Track Sequence - Start / Stop acquiring data from the GPS receiver repeatedly using the given time interval between readings. When this button is pressed in, each specified time interval repeatedly commands data acquisition and the creation of objects or moving map action as specified by the Track Center/Line/Points/Fields buttons.

Download Waypoints - Enabled only when connected to a Garmin device using Garmin protocol or when an NMEA device is detected as sending waypoints. Press to download waypoints. Press again to cancel downloading. When pressed, the system will begin downloading waypoints from the GPS receiver, arranging them in order and creating a point object for each waypoint. A line object will be created for a route. To cancel the download process without creating any objects, click the Download / Waypoints button again before the download is complete.

See the Waypoints and Projections paragraph below for important notes on projections.

Upload Waypoints - Enabled only when connected to a Garmin device using Garmin protocol. Press to upload waypoints. Press again to cancel uploading.

Properties - Set up the serial port connection to the GPS receiver by specifying either a USB connection or the serial port used and the serial protocol options required by the GPS receiver.

Drawing Choose the drawing into which objects and data acquired
from the GPS will be placed.

**Refresh** Specify the time interval and time units (minutes, milliseconds or seconds) between automatic data acquisition when tracking.

**Datum** Datum used by the GPS receiver. The datum setting dialog is resizable.

**Altitude** Value reported by the GPS receiver for the height of the GPS antenna. Although handheld GPS receivers integrate the antenna into the instrument case, remember when using an external antenna the GPS device reports the position of the antenna.

**Bearing** True heading computed by the GPS receiver for its motion based on differences between recent positions measured. The true heading is the magnetic heading corrected for magnetic variation reported for the current position by the GPS device.

**Mag bearing** Magnetic bearing.

**Mag variation** Magnetic variation.

**Latitude** Current position latitude.

**Longitude** Current position longitude.

**Quality** Circular error probability computed by the GPS receiver for the position it is reporting.

**Speed** Speed in centimeters per hour computed by the GPS receiver for its motion based on differences between recent positions measured.

**Satellites** The number of GPS satellites in view of the GPS receiver.

**(status bar)**

### Properties Dialog

**Device** Serial port or USB device to which the GPS receiver is connected. Most computers support COM1 and COM2 with connectors on the chassis. Manifold will automatically search the first 1000 possible COM port configurations and allow use of any COM port that appears to be working. If a USB port is used choose either the Garmin USB connection or the generic USB Human Interface connection if a non-Garmin USB GPS is in use. USB choices will not be available if a USB GPS device is not detected.

**Baud rate** Speed of the serial connection. Use 4800 for most GPS receivers.

**Data bits** Number of data bits in the serial protocol. Use 8 for most GPS receivers.

**Stop bits** Number of stop bits in the serial protocol. Use 1 for most GPS receivers.

**Parity** Parity setting for the serial protocol. Use None for most GPS receivers.

**Test** Test connection to the GPS receiver. Instructs Manifold to listen for valid NMEA sentences.
Consult your GPS receiver’s user manual for information on what settings to use for the \textit{Properties} dialog when using a serial port connection. The \textit{Properties} dialog will fail to launch if there are no available COM ports or USB devices available. Unplugging a USB GPS device will automatically disconnect the GPS Console.

Pressing the \textit{Test} button in the \textit{Properties} dialog commands Manifold to try connecting to the GPS to see if it is alive and well and to report, if possible, what protocol the GPS recognizes. If a GPS is found, a message box will pop open to report whether it is using NMEA or GARMIN protocol.

If the GPS uses NMEA protocol, Manifold will report the NMEA tags recognized.

If the GPS uses GARMIN protocol Manifold will report a list of recognized GARMIN subprotocols recognized as well as the version of each subprotocol. If a GPS device switches from GARMIN to NMEA protocol, Manifold will automatically detect the change.

\textbf{GPS Columns Dialog}

The \textbf{GPS Columns} dialog allows specification of NMEA data that will be downloaded from the GPS, if available. Note that not all GPS devices will transmit all of the following fields. For example, only a depth sounder that repeats GPS location information while transmitting depth information as an NMEA repeater will provide \textbf{GPS Depth} information. The complete list of fields in the \textbf{GPS Columns} dialog is taken from all NMEA sentences that the Manifold GPS Console NMEA parser recognizes.
To capture a given field, double click into the Save To cell for that field and choose either the name of an existing column or to create a new column for that field choose [New Column].

**Select All** - Capture all fields.

**Select None** - Capture no fields.

**Select Inverse** - Uncapture all captured fields and capture all uncaptured fields. A fast way to use all but one field: click Select None, capture the one field not desired and then click Select Inverse.

**GPS Air Temperature**  
Air temperature.

**GPS Altitude**  
Altitude.

**GPS Bearing**  
True bearing.

**GPS Depth**  
Depth from the sounding transducer to the sea floor. Note that transducers are normally mounted some distance below the water line of a vessel, so that to compute the depth of water we may need to add the distance between the vessel’s water line and the transducer to the reported GPS Depth figure. Some depth sounders reporting depth in an NMEA stream have a configuration option which allows adding a fixed distance between the water line and the transducer so that the GPS Depth reported will be the distance from the water line and not from the transducer.

**GPS Dew Point**  
Dew point.

**GPS Humidity**  
Humidity.

**GPS Humidity, Relative**  
Relative humidity.

**GPS Latitude**  
Latitude of current position.

**GPS Longitude**  
Longitude of current position.

**GPS Magnetic Bearing**  
Magnetic bearing (computed from true bearing using variation).

**GPS Magnetic Variation**  
Variation used by the GPS for this position and
<table>
<thead>
<tr>
<th>GPS Quality</th>
<th>Estimated error of GPS position.</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPS Receiving Frequency</td>
<td>Receiving frequency.</td>
</tr>
<tr>
<td>GPS Satellites</td>
<td>Number of satellites used by the GPS to compute position.</td>
</tr>
<tr>
<td>GPS Speed</td>
<td>Speed of motion.</td>
</tr>
<tr>
<td>GPS Stream Bearing</td>
<td>Stream bearing.</td>
</tr>
<tr>
<td>GPS Stream Magnetic Bearing</td>
<td>Stream magnetic bearing.</td>
</tr>
<tr>
<td>GPS Stream Speed</td>
<td>Stream speed.</td>
</tr>
<tr>
<td>GPS Time</td>
<td>Time of measurement in Universal (Zulu) time.</td>
</tr>
<tr>
<td>GPS Time, Local</td>
<td>Local time</td>
</tr>
<tr>
<td>GPS Time Zone, Local</td>
<td>Local time zone.</td>
</tr>
<tr>
<td>GPS Transducer to Keel</td>
<td>Distance from the depth sounding transducer to the lowest point of the vessel. Transducers are normally mounted below the waterline but significantly above the keel. The distance from the transducer to the sea bottom must exceed this value to avoid grounding the vessel.</td>
</tr>
<tr>
<td>GPS Transducer to Water Line</td>
<td>Distance from the depth sounding transducer to the water line of the vessel. Transducers are normally mounted below the waterline but significantly above the keel. This distance should be added to the depth from the transducer to the sea bottom to get water depth.</td>
</tr>
<tr>
<td>GPS Transmitting Frequency</td>
<td>Transmitting frequency.</td>
</tr>
<tr>
<td>GPS Water Temperature</td>
<td>Water temperature.</td>
</tr>
<tr>
<td>GPS Waypoint Name</td>
<td>Used when downloading waypoint information.</td>
</tr>
<tr>
<td>GPS Wind Bearing</td>
<td>Absolute bearing of the wind.</td>
</tr>
<tr>
<td>GPS Wind Bearing, Relative</td>
<td>Apparent wind bearing considering the bearing of the vessel.</td>
</tr>
<tr>
<td>GPS Wind Speed</td>
<td>Absolute wind speed.</td>
</tr>
</tbody>
</table>

The availability of numerous additional columns, such as GPS Depth, opens the door to numerous applications if we have the equipment that can provide such NMEA information. For example, suppose our boat has a depth sounder that can read GPS latitude / longitude information from a GPS device (either external or built-in) and can then repeat it on an NMEA serial connection as an NMEA "talker" along with depth information generated by the sounder.

We can then automatically map depths for a water body by simply connecting the NMEA serial cable to a laptop running Manifold and then set the GPS console to record GPS Depth as well as GPS Latitude and GPS Longitude. We set Track Columns and Track Points on and then cruise around the body of water for all regions where we want to capture depths. We could use the resulting depths to create a surface that shows the bathymetry of the water body, or we could create labels from the depth values to create a chart with traditional depth sounding values in the chart. If we need to make any adjustments to the depth values captured, such as, for example, a correction for the distance between the water line and the transducer we could use the transform toolbar for tables to add a fixed value to all of the depths.

**Protocol Requirements for Downloading and Uploading Waypoints**

The Upload Waypoints command works only with Garmin devices connected using Garmin protocol that, further, support the relevant Garmin subprotocol. Some older Garmin devices do not support Garmin uploading protocol.
Although the **Download Waypoints** command works when either NMEA or Garmin protocols are used for connection, as a practical matter it is easiest to use with Garmin protocol. When used with Garmin devices in Garmin protocol, the **Download Waypoints** button is enabled upon connecting to the GPS device and one simply presses the button to download whatever waypoints are stored in the GPS device.

When using NMEA protocol, the **Download Waypoints** command will stay disabled until the GPS device actually transmits some waypoints. Many units using NMEA protocol do not transmit waypoints unless specifically instructed to do so via the devices user interface. For example, the device may only be able to send waypoints for download if they are part of a route and that route has been made “active.”

It is such a hassle to decode the user documentation for most GPS devices to figure out what must be done to send waypoints down the pipe for download, and such an annoyance to fumble with thumb-based menus that most people who want to download waypoints from a GPS will simply buy an inexpensive Garmin device and use that with Garmin protocol.

Note that as in the case for uploading waypoints, when using a Garmin device it must be a reasonably modern device that supports the required Garmin download subprotocol.

**Downloading Waypoints, Routes and Tracks**

Most GPS devices can store **waypoints**, which are point locations noted by the GPS. Most GPS devices make it easy to mark a waypoint at a desired location. In addition, most GPS devices also allow designation of **routes** and acquisition of **tracks**. A route is usually a list of waypoints. A **track** is a series of positions recorded by a GPS device and is imported into Manifold as a line. While routes are formed from waypoints, tracks are entities independent of waypoints. Manifold can import waypoints, routes and tracks.

Download of waypoints, routes and tracks and upload of waypoints in Manifold works only with Garmin protocol, which means we must be using a GPS device manufactured by Garmin. Garmin units implement two different protocols for downloading routes and tracks, so Manifold provides two separate checkboxes, one for downloading routes and the other for downloading tracks. If a particular Garmin unit does not implement either the route or track download protocol, the relevant checkbox will be disabled.

**To acquire waypoints from a GPS device:**

1. Attach a Garmin protocol GPS receiver to a serial port or USB port on the computer. Note which serial port (COM1, COM2, etc) is used.
2. Launch Manifold. Open the project into which the GPS Console will save objects. Create a drawing and verify the projection of that drawing using **Assign Projection** if a drawing does not yet exist. Using **Edit - Change Projection**, change the projection of the drawing to **Latitude / Longitude**.
3. Open the GPS pane with a **SHIFT-ALT-G** or with a **View - Panes - GPS Console** command from the main menu.
4. In the GPS Console, choose the drawing into which objects will be saved. The drawing cannot be a read-only drawing (such as a read-only drawing on an Enterprise server in Enterprise Edition) and it must be in **Latitude / Longitude** projection.
5. Press **Properties** in the GPS Console toolbar. If a serial port is used, choose the serial port and the serial port properties for the GPS device. Most GPS devices use the default baud rate of **4800** and other default settings. If a USB port is used, choose either the Garmin USB connection or the generic USB Human Interface connection if a non-Garmin USB GPS is in use. Press **OK**.
6. Press the **Download Waypoints** button in the GPS console toolbar.
7. In the resultant **Download Waypoints** dialog, choose options desired such as whether to download waypoints, routes and tracks, which data fields should be downloaded and into which columns should they be imported. Press **OK**.
The **Download Waypoints** dialog provides the following controls:

- **Download waypoints** Check to download waypoints.
- **Download routes** Check to download routes.
- **Download tracks** Check to download tracks.
- **Reuse downloaded data** Recycle already downloaded data instead of fetching it again from the GPS device. Used to save already downloaded data into different drawings.
- **Create new drawing** Create a new drawing, called **GPS Data** by default.
  - **Choose all** - Download all columns.
  - **Choose none** - Download no columns.
  - **Choose inverse** - Reverse values in **Save To** column so that those columns currently marked for download are now downloaded and vice versa.

**Column** Columns available for download.

**Save To** Columns into which downloaded data should be placed. Click into the **Save To** cells to choose from pull down menus of columns in the destination drawing's table. Choose [New Column] to create a new column in the drawing to accept the data.

Waypoints in a GPS normally have a latitude, longitude and altitude for each waypoint. In addition, each waypoint has a name and a symbol used to represent that point in the GPS. Some waypoints may also have a comment in the form of a short text string.

Once download occurs, the GPS Console will report the results of the download in the status bar at the bottom of the display.
Note that because GPS connections can be very slow the download of tracks can take a long time. Manifold will provide progress indications during the download process.

Due to the slow interface used by GPS devices the waypoint download process can be very slow at times, with only a few waypoints per second being downloaded. Tracks especially can take a long time. Manifold will provide progress indications during the download process.

When creating a new drawing using the Create new drawing option the new drawing will be created using Latitude / Longitude instead of the usual default Orthographic projection.

**Reusing Downloaded Data**

The GPS Console will cache downloaded data so that unnecessary downloads can be avoided if possible. The Reuse downloaded data option tells the GPS Console to recycle any data already downloaded instead of freshly downloading it. This option will be enabled only if there is some data available to reuse.

Use this option if you have downloaded data from a GPS device and there have not been any changes made to the data in the device, for example, no new waypoints added. If we have just downloaded data into a drawing and we would also like to download those points into a different drawing we can use this option to avoid the slow download process.

It is true that we could simply copy the downloaded points from the first drawing into the second, but it could be that the drawing into which the waypoints were first downloaded already had many points in it and that it is easier for us to simply download the waypoints again into a second drawing than it is to figure out which of the many points in the first drawing are the recently downloaded waypoints that should be copied.

To reuse downloaded data:

1. Connect to a GPS unit using Garmin protocol.
2. Select a target drawing in the Drawing box.
3. Download data using the Download Waypoints toolbar button.
4. Select another target drawing in the Drawing box.
5. Press the Download Waypoints button again.
6. Check the Reuse downloaded data option and click OK.

**Waypoints and Projections**

Downloaded waypoints will not be created if doing so would create an object outside the region that the projection in use can show. This prevents the addition of data that is impossible to show in the projection in use.
For example, the Orthographic projection used by default for new, blank drawings can only show one hemisphere of the Earth at a time. Waypoints that would appear in the hidden hemisphere of the default Orthographic projection will not be created. For this reasons, drawings created automatically when downloading waypoints will be created in Latitude / Longitude so waypoints will always be created no matter where they are located on Earth.

To avoid needing to think about projections the simplest approach to deal with the above is to create a new drawing that will accept the waypoints and then to immediately use Edit - Change Projection to change the projection of the drawing from the default Orthographic into Latitude / Longitude. There are no hidden hemispheres or disallowed locations in Latitude / Longitude so waypoints will always be created no matter where they are located on Earth.

Users who have confidence in their ability to use projections correctly can create a new drawing that uses a projection appropriate to their area of interest in which the waypoints will appear. The easiest way to do this is to create a map based upon a drawing that shows reasonable background features (such as roads or cities) for the area of interest and to then add a new drawing to the map by right clicking a layer tab in the map and choosing Add - New Drawing. The new drawing will automatically be created in the projection used by the map, which, if the map shows the area of interest correctly, will of course be a reasonable projection to use.

Adding a new drawing to an existing map to accept waypoints is also a good idea because seeing the waypoints appear against a background map that provides visual context is a good check against errors. For example, if we are working with a Garmin device and downloaded waypoints appear in the expected places we know we’ve chosen the correct route’s waypoints (if there is more than one route full of waypoints stored on the GPS device) for download.

### Uploading Waypoints

Many GPS device can receive uploaded points and store them as waypoints for later use. Such waypoints usually may be displayed in the GPS device’s built-in maps (if any) and used for navigational purposes.

Manifold can upload points from a drawing into a GPS device as waypoints if the following conditions are observed:

- Uploads require a Garmin device connected using Garmin protocol.
- The drawing from which uploads occur can only contain points.
- The drawing must be in Latitude / Longitude projection.
- The drawing’s table must contain a column that contains a unique, short name for each point.
- The GPS device must be capable of accepting uploads of waypoints. Note that some Garmin devices, such as the “hockey puck” simplified receivers like the GPS18 and later such devices, cannot store waypoints within the device nor accept uploads of waypoints.

To upload waypoints into a GPS device:

1. Attach a Garmin protocol GPS device capable of accepting waypoints to a serial port or USB port on the computer. Note which serial port (COM1, COM2, etc) is used.
2. Launch Manifold. Open the project into which the GPS Console will save objects. Make sure the drawing from which points will be uploaded is in Latitude / Longitude.
3. Open the GPS pane with a SHIFT-ALT-G or with a View - Panes - GPS Console command from the main menu.
4. In the GPS Console, choose the drawing from which points will be uploaded.
5. Press Properties in the GPS Console toolbar. If a serial port is used, choose the serial port and the serial port properties for the GPS device. Most GPS devices use the default baud rate of 4800 and other default settings. If a USB port is used, choose either the Garmin USB connection or the generic USB Human Interface connection if a non-Garmin USB GPS is in use. Press OK.
6. Press the Upload Waypoints button in the GPS console toolbar.
7. In the resultant Upload Waypoints dialog, choose the column that will be used to provide the GPS Waypoint Name value for each point and, optionally, the columns that will provide the GPS Waypoint Symbol and GPS Waypoint Comment values if such are desired. Press OK.

### Upload Waypoints Dialog
The Upload Waypoints dialog provides the following controls:

<table>
<thead>
<tr>
<th>Column</th>
<th>Load From</th>
</tr>
</thead>
<tbody>
<tr>
<td>[GPS Waypoint Comment]</td>
<td>Comments</td>
</tr>
<tr>
<td>[GPS Waypoint Name]</td>
<td>Name</td>
</tr>
<tr>
<td>[GPS Waypoint Symbol]</td>
<td>Symbol</td>
</tr>
</tbody>
</table>

Note that each point's name should be unique in whatever column is used to provide the GPS Waypoint Name. Using the same name for a subsequent point will simply overwrite any previous waypoint by that name.

Examples

The following examples assume the GPS receiver has been correctly connected to the computer and the GPS Console has been set up.

To connect the GPS receiver and set up the GPS Console:

1. Attach an NMEA or Garmin compatible GPS receiver to a serial port or a USB port on the computer. Note which serial port (COM1, COM2, etc) is used.
2. Launch Manifold. Open the project into which the GPS Console will save objects.
3. Open the GPS pane with a SHIFT-ALT-G or with a View - Panes - GPS Console command from the main menu.
4. Press Properties in the GPS Console toolbar. Configure USB port or serial port properties for the GPS device. Press OK.
5. Press Connect / Disconnect in to connect to the GPS device.

When working with GPS devices we will often travel from position to position. We would like to use our current position as reported by the GPS to move a map so that our position is always at the center of the map.

Move a map to show GPS location at the center:

1. Open the map desired.
2. Choose a drawing in the map as the Drawing choice in the GPS Console. Connect to the GPS device by pushing in the Connect / Disconnect button.
3. Choose a reasonable refresh interval, such as every 5 or 10 seconds. What is reasonable depends on the travel speed and the scale of the map. There is no point in refreshing the display every second if one's travel speed is so slow or the scale of the map so big that it takes minutes for any change in position to be noticeable.
4. Choose (push in) Track Center in the GPS Console toolbar.
5. Press IN Track Sequence in the GPS Console toolbar.

We might want to show a reticule at the center of the map when in Track Center, "moving map," mode. Reticules are controlled from the World pane.
Show a reticule at the center of the map:

1. Open the subject map.
2. Open the World pane.

A very common use of GPS Console is to create maps by driving the periphery of ranches or over roads in a vehicle equipped with a GPS receiver and a laptop running Manifold. The GPS Console is set up to record position by creating points over a regular time interval.

To create a series of points at a given time interval:

1. Open the subject map or drawing in which points will be created.
2. Choose the drawing as the Drawing choice in the GPS Console. Connect to the GPS device by pushing in the Connect / Disconnect button.
3. Choose a reasonable refresh interval, such as every 5 or 10 seconds. What is reasonable depends on the travel speed, the scale of the map and the desired accuracy.
5. [Optional] In the Track Columns dialog, choose [New Column] as the Save To method or all fields to be imported. Choose a blank value in the Save To method to not import a field. If mapping a parcel of land using higher accuracy GPS receivers many people would import GPS Altitude.
6. Choose (push in) Track Points in the GPS Console toolbar.
7. Press IN Track Sequence in the GPS Console toolbar. The system will begin creating points.
8. Press OUT Track Sequence in the GPS Console toolbar to stop creating points.

Note that the act of creating a point in a drawing from a GPS position automatically captures the latitude and longitude of that point. Even if we do not import the GPS Latitude and GPS Longitude fields from the GPS device at that instant using the Track Columns option, we always can see the latitude and longitude of that point by showing the Latitude (I) and Longitude (I) intrinsic fields in the drawing’s table.

As an option we may wish to capture fields from the GPS that are not implied by the location of the points created. For example, we may wish to capture GPS Altitude so that each point has a data field giving the altitude of that point.

Example Scenario

We work for a rural township and would like to create a precise map of all addresses on rural routes. We dispatch a vehicle with a driver and a person to operate a laptop computer connected to a GPS device. They will capture a point for each address as follows:

1. Configure Manifold using a drawing that shows streets in the township. In the map, add a blank drawing called Points to receive GPS-generated points.
2. Click open the table associated with the Points drawing and create an Address field that is a variable length text field.
3. Configure the GPS console to save points to the Points drawing. Press in Track Center and Track Points and configure Track Columns to save Latitude and Longitude. Connect to the GPS device by pushing in the Connect / Disconnect button.
4. Enable Edit - Instant Data .
5. Drive to each address, pausing at the address.
6. At each address, press the Track Once button. This will pan the map to that location, save a point at that location and will raise the Instant Data dialog.
7. In the Instant Data dialog, select Address as the field and enter the street address.
8. Continue in this fashion for all addresses, driving to each and pressing the Track Once button. With each future address Instant Data will automatically position the cursor for data entry for the Address field.

When adding points it is helpful to see the results of a data entry as points appear. We can add a labels component to the map so that each point is labeled with the Address field as it is created. To do so, copy the
drawing and paste it as a labels component using the Address field. Drag and drop the new labels component into the map. As each point is created the address for the label will appear.

The result of the above scenario will be a drawing with a point for each address. Each point will have a latitude, longitude and address field. This same process can be used to locate all bridges, fire hydrants, manhole access points, transformers or any other facility. For locations in urban areas, use an WAAS-enabled GPS receiver to achieve accuracy of 3 meters or better.

**Accuracy**

Typical GPS accuracy is approximately 15 meters worldwide. For many years, the U.S. military deliberately damaged the accuracy of signals sent out by satellites to civilian GPS devices so that their accuracy averaged only 100 meters. The signal sabotage program was called “Selective Availability” (SA). SA was turned off by Presidential order in the year 2000 to allow all GPS devices whether military or civilian to operate at full GPS accuracy of 15 meters. Because most GPS devices use chip sets produced by a limited number of vendors, even the cheapest handheld GPS device delivers about the same default accuracy as the most expensive.

An easy way to see the accuracy of a GPS device is to connect it to a laptop computer and then start recording points without moving the GPS device. As points are collected the actual location for each point will vary slightly due to transient changes in what the GPS computes to be its location.

The above plot shows approximately 750 points collected with a Garmin GPS18 USB device at a stationary location. After collecting the points using Track Points we added an Order field to the drawing and then used the transform toolbar to add an arithmetic series to the Order column beginning with 1. This is just a simple way of numbering each point in order from 1 to 750. We then used thematic formatting to color the background color of the points with a spectrum of colors from purple to red so that there is a visual indication of the order in which points were acquired.

The width of the scattered points is about 6 meters (18 feet) and the height about 4 meters (12 feet). The points were acquired with six satellites in view, but given that the Garmin GPS18 uses WAAS it is not spectacularly good accuracy. Presumably, if more satellites were in view (the GPS18 has 12 channels to handle up to 12 satellites at once) or if WAAS reception were better at the test location then accuracy would be improved.

**Enhanced Accuracy**

Two enhancement systems provide GPS accuracy better the default 15-meter GPS accuracy:

- **Differential GPS (DGPS)** - An older technology that requires additional receiving circuitry usually implemented as an external black box. Provides 3-meter to 5-meter accuracy.
- **Wide Area Augmentation System (WAAS)** - A newer system functioning throughout North America that can be implemented using almost the same receiving circuitry already designed for regular GPS.
function. Provides better than 3-meter accuracy 95% of the time. Europe and Japan are developing similar systems.

Many low cost handheld devices now include WAAS functionality built in for just slightly increased prices over non-WAAS products. At this writing it is possible to buy a WAAS-enabled GPS device for just over $100 that can connect to a laptop computer.

When mapping land parcels with GPS, in North America one should use a WAAS enabled GPS receiver if possible to achieve accuracy better than 3 meters over 95% of the time. WAAS also provides this same accuracy in altitude measurements. An accuracy of better than 3 meters is roughly equivalent to the accuracy of TIGER/Line and USGS 1:100,000-scale SDTS/DLG drawings, so it is a good match for much GIS work.

WAAS-enhanced GPS is also a good match for mapping civic infrastructure such as the locations of fire hydrants, 911 addressing, rural geocoding and many other applications.

**Bearings, Magnetic Bearings and Variation**

**Bearings** are radial measurements describing directional movement on the Earth where travel directly towards the North Pole is assigned a value of $0^\circ$, directly East is $90^\circ$ degrees, South is $180^\circ$ degrees and due West is $270^\circ$ degrees. **True** bearings are given based on this system that uses the bearing toward the North Pole as the zero value. The North Pole, in turn, is defined by the spin axis of the Earth. Bearings are also referred to as "headings".

Magnetic compasses do not point towards the North Pole of the Earth. Instead, they point approximately at the North Pole of the Earth's magnetic field, which is usually located hundreds of miles away from the North Pole defined by the spin axis, approximately 700 miles (1140 kilometers) away. In addition, due to fluctuations in the Earth's magnetic field the magnetic North Pole will move over a period of years relative to the true North Pole.

Because the magnetic North Pole is located hundreds of miles from the true North Pole the bearing shown by a compass will not the same as the true bearing. Instead, the compass shows **magnetic** bearing (also known as "compass" bearing). The difference between the magnetic bearing shown by a correctly compensated compass and the true bearing is called **variation**. Variation is different at different locations on Earth because different locations have a different geometric relationship between the location and the placement of the true North Pole and the magnetic North Pole. Other factors, such as magnetic anomalies in a particular region, can greatly influence variation as well.

Variation can be substantial. For example, the variation in the San Francisco Bay region is approximately $15.6^\circ$ degrees. That means to head directly North one must follow a magnetic bearing of approximately $344^\circ$ degrees as indicated by a compass or as indicated by a GPS device set to display magnetic bearings. Variation can change surprisingly fast in some regions of the Earth. In far north regions or in locations near magnetic anomalies positions that are only a mile (1.6 km) apart can have an entire degree difference in variation.

Variation also changes over time as the Earth's magnetic field drifts to slightly different locations relative to the Earth's spin axis. The changes are quite rapid compared to most geophysical processes; there are locations on Earth where significant (greater than 1%) changes in variation will occur within a few years.

Consumer GPS devices are intended to assist navigation in settings such as boating or hiking where magnetic compasses will also be used. Such GPS devices are often setup by default to display magnetic bearings even though the GPS computation within the unit provides a true bearing. Most GPS devices calculate the magnetic bearing they display by consulting a look-up table within the unit that contains an approximate variation for various locations on Earth. Since the unit knows its location and the current date, it can find the current variation for that location in the table (interpolating if need be) and adjust the true bearing to the magnetic bearing that is displayed.

Garmin, for example, at the present writing uses an internal lookup table based on the IGRF 90 international standard Earth magnetic field model to estimate the variation at any given location. The model as used by Garmin provides +/- 2% accuracy for more than 95% of the Earth's surface through the year 2010. Because variation changes more rapidly in some locations (such as northern Canada) where there is a high variation to begin with and a high annual rate of change in variation, in some locations the internal lookup table will result in accuracy worse than 2% by the year 2010. To deal with such cases Garmin provides the ability to manually specify the variation to be used.

Most units will normally report true bearings via their NMEA interface. They will also report the variation used for that location at the time of the measurement. When capturing data from a GPS it is best to capture the true bearings. The true bearing is an invariant quantity that is linked to the spin axis of the Earth. For tactical
navigation purposes it is convenient to use magnetic bearings with GPS devices so that the bearing reported (and captured within Manifold) is the same as that seen on a compass. However, for archival purposes it is risky to use magnetic bearings because the actual direction indicated by a magnetic bearing depends upon variation, which in turn depends upon the location, the geophysical model employed and the date of the measurement. To maintain archival integrity over the years it is best to capture variation together with magnetic bearings if magnetic bearings will be used.

**Note:** A more precise term for variation is magnetic **declination**. "Variation" is a less technical term employed in marine and air navigation. We use the term **variation** because it is widely understood by lay audiences to be the difference between true bearings and magnetic bearings. A further desire is to avoid confusion with the meaning of "declination" in celestial navigation and astronomy where together with right ascension it is used to describe the position of a star.

**Datums**

The default datum used by almost all GPS devices is WGS84. Manifold assumes data coming in from the GPS receiver is in latitude / longitude coordinates using the WGS84 datum. When GPS information is added to a drawing, the GPS coordinates will be converted on the fly from latitude / longitude coordinates and WGS84 into whatever coordinate system (projection) and datum are used by the drawing.

If your GPS receiver is set to use some other coordinate system (projection) or some other datum, then Manifold's interpretation of the data it receives from the GPS receiver will not be correct. The easiest way to avoid such problems is to make sure your GPS receiver is set to use WGS84 and latitude / longitude coordinates. If your GPS receiver is set to use other coordinates or a different datum, the easiest path is to use the setup menu in your GPS receiver (see the documentation for your GPS receiver) to change it to use latitude / longitude and WGS84.

**Troubleshooting**

If your system is unable to read data from the GPS receiver check the following (using a log file if necessary):

- Is your GPS turned on and transmitting location data? Some GPS devices do not transmit location data until they have acquired a signal.
- Is your GPS a serial port interface device or a USB interface device?
- Is the GPS device plugged into your computer's serial port, or, if it is a USB device is it plugged into the USB port?
- Is your GPS interface set to NMEA version 2? Most modern GPS receivers when set to "NMEA" for their interface will automatically use version 2. Some GPS devices, such as Garmin, may come with their own proprietary interface protocol selected by default. Although Manifold can communicate with Garmin devices in Garmin protocol, for other proprietary formats you will have to change the interface protocol to NMEA version 2 in the GPS receiver's setup menu. If you experience any difficulty acquiring fields when working with Garmin protocol, set the device to use NMEA protocol.
- Is your serial port or USB port on your computer functioning correctly?
- Have you chosen the correct serial port in the **Properties** dialog? If using a USB port have you specified a USB connection?
- If you are using a manufacturer's cable to connect a serial port GPS device to a USB connection and that manufacturer provided custom USB drivers, did you install the USB drivers successfully?
- When using a serial port connection, do the baud rate and other serial port settings in use for the computer match the settings used by the GPS device?
- Some GPS devices will not begin transmitting until an active route is specified or some other setup procedure is followed. If this is the case with your device have you correctly set up the GPS?
- Use HyperTerminal to manually check the GPS interface as described below.
- When downloading waypoints from some Garmin devices, the route containing the waypoints must be set as the **active** route. See your GPS device user manual for information on how to designate a particular route as the active route.

If waypoints do not appear when you download them, make sure the drawing you are using is in **Latitude / Longitude** and not some other projection.

Details on GPS receiver interfaces and setup, by long tradition in the GPS hardware industry, are buried in a very obscure portion of the GPS receiver's user manual. It may well be a frustrating exercise to try and find out how to set a route active, how to configure your device to use NMEA 2 and so forth. For example, the manual for a recent Garmin device talks about waypoints and "active" routes but there is no command or documentation
for designating an active route; instead, the user must infer that telling the unit to “navigate” a particular route makes it the active route.

Do an Internet search for pages with advice, tips and tricks on using GPS receivers.

**Track Sequence Timing**

The *Track Sequence* command might not operate continuously if CPU usage is too high. The *Track Sequence* command operates when the system has nothing else to do (to avoid crowding the drawing with too much data) so it may get overridden in extremely heavily loaded systems by higher priority processes. The solution is to avoid running too many demanding programs that place heavy loads on the system processor at the same time while running Manifold with the GPS console and *Track Sequence*.

**Garmin Protocol**

Garmin GPS devices can communicate with using either industry-standard NMEA protocol or using Garmin's own proprietary interface protocol. Manifold can connect to a Garmin GPS device using either NMEA protocol or Garmin protocol.

**Technical Note on NMEA Parsing**

The GPS Console handles NMEA sentences in a generic manner. For example, the handler for the $GPGGA sentence can handle arbitrary $xxGGA sentences. Manifold can handle more than 20 NMEA sentences, including DBT, DPT, FSI, GDA, GDF, GDP, GGA, GLA, GLF, GLL, GLP, GOA, GOF, GOP, GSA, GXF, GXP, HCC, HDG, HSC, MHU, MTA, MTW, MWV, Rnn (where nn is a decimal number), RMA, RMC, RMZ, VDR, VHW, VTG, WPL, ZDA, ZLZ, ZZU. Not all are literally exposed as available fields in the *GPS Columns* dialog, but all will be used as necessary to extract the fields available in the dialog.

Some devices attempt to extend NMEA through the use of proprietary sentences that are unique to that manufacturer. Manifold does not support proprietary NMEA sentences.

**Choosing a GPS Device**

There are hundreds of different brands of GPS receivers that work perfectly well using standard, non-proprietary NMEA format; however, on occasion you may encounter some devices that claim to be "NMEA compatible" that fail to communicate via standard NMEA sentences.

Some such devices simply have bugs in their firmware while others inject proprietary sentences into what is otherwise a standard NMEA data stream. Devices that mix proprietary sentences with standard NMEA sentences can be perversely confusing to new users because they may work for some features, such as reporting latitude and longitude, but not work for other features, such as downloading waypoints. Even well-known brands, such as Magellan, may incorporate such communications incompatibilities into their NMEA interfaces, so users should not take it for granted that an "NMEA compatible" device really is NMEA compatible.

Another source of incompatibility may occur when using devices that connect via USB. This especially may be the case when working with USB devices using vendor-supplied USB drivers, in which case not only might GPS communications standards not be observed but a variety of non-standard approaches to providing Windows drivers might also be a factor.

**Manifold.net** recommends Garmin GPS devices because they are high quality, inexpensive, very reliable and do an excellent job of supporting standard industry protocols. In addition, Garmin has posted for free download all information necessary for software vendors to do a good job of supporting Garmin devices both in NMEA and in proprietary Garmin mode. Garmin also normally provides even many of its least expensive devices with full capability to connect with and interact with PCs. This makes it possible for users to acquire a Garmin device and connect it to their PC at a very low cost. Garmin devices make it easy to download waypoints, routes and tracks and are the only brand of GPS devices currently supported by Manifold for uploading waypoints.

Because of the high quality of Garmin firmware and support for standard protocols, **manifold.net** uses Garmin devices as the "gold standard" for GPS compatibility in NMEA and Garmin interfaces. If a Manifold feature does not work with a Garmin device, it is immediately investigated as a possible bug in Manifold. In contrast, if a Manifold feature works with the Garmin devices in our testing laboratories but is reported not to work with some other GPS manufacturer's device, the debugging process will begin with the assumption that the problem is in that other GPS manufacturer's firmware or communications protocol.
If you have a problem using a GPS device other than a Garmin device, Manifold technical support (at the cost of one or more support incidents) is always willing to investigate the matter for you to help verify that all settings (such as baud rate, etc.) are correct. However, beyond the initial stages of investigation good documentation from the GPS manufacturer may be required for resolution of the problem. Absent such documentation, if tests with a Garmin device do not duplicate the problem the fault will be assumed to be a problem in the GPS device you are using and not in Manifold.

**Support for Proprietary GPS Features**

*manifold.net* is able to provide the best support for those GPS vendors, such as Garmin, who do a good job of providing technical details for their devices for free download on their websites. We are always willing to listen to requests for support of proprietary GPS protocols and features, but only if the GPS manufacturer publishes complete details required for such support for free download and free usage.

When vendors publish on their websites details of formats for communicating with and controlling their devices, for uploading and downloading waypoints, for uploading cartography and so on, it makes it possible for software developers such as *manifold.net* to provide crisp and reliable support for those devices. We believe this minimizes the tech support load on both *manifold.net* and the GPS vendor while it increases customer satisfaction and makes it more likely that the GPS vendor will sell more devices.

There are now hundreds of GPS manufacturers, some of which are not really GPS developers but are more accurately understood to be integrators, in that they take an OEM version of a GPS chipset or module made by some other company and simply package it, relying upon the manufacturer of the module to provide software support. Some such integrators do not provide good support for their products because they do not understand the internal functioning of the modules and software they buy from the original developer.

If your GPS vendor does not provide good documentation online, consider switching to a vendor who does. Given the very low cost of GPS devices and the great competition in this market there is little reason to do business with vendors who do not provide full and precise documentation online.

See the Contacting manifold.net topic if you would like to recommend support for proprietary GPS features.

**GPS Manufacturers**

If you are a bona fide GPS manufacturer, *manifold.net* technical support will provide free Manifold System licenses to you so you can test your GPS devices with Manifold and better support your customers. This offer is open only to manufacturers and not to dealers, distributors, integrators or other non-manufacturing entities.

**Technical Note on Datums**

If your GPS receiver uses some coordinate system and datum other than latitude / longitude and WGS84 and cannot be conveniently reset, it is still possible to use data from that GPS receiver. To do so, first create a drawing in latitude / longitude and WGS84 and acquire the GPS data into that drawing. Next, use the Edit - Assign Projection dialog to specify the coordinate system and datum used by the GPS receiver for that drawing.

See the Technical Note for Experts at the end of the Projections topic for variations in WGS84 and NAD83 datums as used by certain GPS devices.

**See Also**

- View - Panes
- View - Panes - GPS Console
### Key Windows and Dialogs

This topic summarizes the most frequently used windows, panes and dialogs in Manifold. See the individual topics for each window or dialog for details.

#### Windows

<table>
<thead>
<tr>
<th>Label</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maps</td>
<td>Map windows are the main user interface in Manifold, because they allow use of many layers.</td>
</tr>
<tr>
<td>Tables</td>
<td>Tables show database table contents and provide a row and column user interface to database information.</td>
</tr>
<tr>
<td>Charts</td>
<td>Charts provide visual summaries of information and a visual framework for data mining through click-back selection.</td>
</tr>
<tr>
<td>Drawings</td>
<td>Drawings are usually opened as layers within a map window. Drawing windows are only used if the native coordinates of the drawing must be re-projected.</td>
</tr>
<tr>
<td>Themes</td>
<td>Themes are formatted drawings. They are a convenient way of showing the same drawing using different formatting choices.</td>
</tr>
<tr>
<td>Images</td>
<td>Images are usually opened as layers within a map window. Image windows are only used if the native coordinates of the image must be re-projected.</td>
</tr>
<tr>
<td>Surfaces and Terrains</td>
<td>Surfaces are 2D presentations of raster data that normally shows terrain elevation data, but which can be any continuously-varying data. Terrains are a 3D view of a surface. If we have Enterprise Edition we also get Profiles and Elevations, which are &quot;cuts&quot; through a surface.</td>
</tr>
<tr>
<td>Forms</td>
<td>Form windows are the visual interface to forms-based programming of scripts as well as to form creation for work with databases.</td>
</tr>
<tr>
<td>Queries</td>
<td>Queries allow creation of SQL queries for selection and manipulation of database information. Results are displayed as tables.</td>
</tr>
</tbody>
</table>

#### Panes

<table>
<thead>
<tr>
<th>Label</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project</td>
<td>Used to import, insert and organize components.</td>
</tr>
<tr>
<td>Control Points</td>
<td>Control points are used georegister images and drawings. The Control Points pane provides a listing of control points defined in the current image or drawing.</td>
</tr>
<tr>
<td>GPS Console</td>
<td>The GPS Console pane controls acquisition of data from a GPS receiver connected to your computer via a serial port. The GPS Console can automatically move a drawing or map in &quot;moving map&quot; mode as well as fetching data from the GPS on demand or automatically on a regular time interval.</td>
</tr>
<tr>
<td>Info pane</td>
<td>The info pane shows data attributes for objects one record at a time. It is a convenient way of seeing data from drawing tables one record/object at a time, by selecting objects or by stepping data one object at a time.</td>
</tr>
<tr>
<td>Layers</td>
<td>Displays the layer stack in a map. The layers pane is more convenient to use with many layers than the tab controls at the bottom of the map. When used with image windows, the layers pane turns individual R, G, B or alpha channels on and off and also controls whether the image border or background is</td>
</tr>
</tbody>
</table>
displayed.

**Review**

Use with Enterprise Edition and above with concurrent, multi-user editing of linked drawings.

**Selections**

Provides a scratchpad for saving selections that have been made so that they may later be re-used.

The selection pane is very important when working with maps because saved selections are used extensively with the transform toolbar.

**Tool Properties**

Displays settings for the current tool in use, for example, threshold for Select Touch when selecting with images, or the opacity of a paintbrush when editing images.

**ViewBots**

ViewBots are one-line analytic instruments that report a result or otherwise make a calculation using the contents of a window.

**World**

The World pane shows current position and controls Linked Views.

**View Menu**

**View - Properties**

Used to set custom background color, precision factor, zoom ranges and the projection properties of a component. Zoom Ranges control the zoom levels over which a component will be visible in a map, the intrinsic projection properties and the precision parameter.

**View - Properties - Link / Share Status**

Used to see connection parameters and edit modes of linked components and shared components. Essential when managing multi-user editing of linked drawings or using shared components with Enterprise Edition.

**View - Properties - Precision**

The Precision dialog sets a location precision distance that defines the radius of action of many commands and operators, including SQL. For example, some drawing transform toolbar operators like Normalize Topology will snap a line end to a point if the line end is closer to the point than the distance given in the Precision dialog.

This is a critically important parameter that must be understood when using Normalize Topology or other advanced commands.

**File Menu**

**Export - Web Page**

Creates a web page that can be published to the Internet via Internet Information Server. See the Map Server Overview topic.

**Tools Menu**

**Make Image**

Save the open component (map, drawing, surface, etc) as an image. A highly useful way of merging two images into one image. Also used to quickly save a map as an image that can be used in other applications (for example, for use in Microsoft Word or on a Web page). When used to save a print Layout as an image, can save the image to any resolution desired.

**Administrator Console**

Appears only in Database Administrator editions of Manifold. Provides key managerial and user-interface functions which the administrator can use to set up the
database for greater ease of use and capability for all users.

Database Console

Very frequently used to connect to external databases, to browse them and to link tables or other components into Manifold. The number one way of linking components from Oracle or other databases for multi-user editing using Enterprise Edition.

Edit Menu

Edit - Change Projection

Used with components like drawings and images to re-project the component and thus permanently change the data.

Edit - Assign Projection

Used with drawing, image, labels and surface windows to change the interpretation of the data. This is a specialized function that is applied only when manually specifying the projection of a component imported from a format that does not correctly store projection information. To change the native projection of any component other than a map, use the Edit - Change Projection command.

Panes that are regularly left open, like Selections, Layers, Info and ViewBots will automatically switch context as the focus switches from one window to another window.

Panes may be docked into the main Manifold window or left undocked. See the View - Panes topic for more information on panes. Manifold makes heavy use of panes.

The Status Bar

The status bar at the bottom of the screen displays information on command status, location and other useful information. The leftmost region of the status bar is used to display brief descriptive phrases or captions relevant to the current command, somewhat like extended tool tips. Not all commands will show such descriptive phrases.

The rest of the status bar provides readouts that apply to the currently active window. Not all readouts will be active for all types of windows. For example, the Location and Scale readouts will not be active for table windows. The Scale readout will be active only for projected components and maps.

<table>
<thead>
<tr>
<th>Orthographic</th>
<th>-108.8484 25.4957</th>
<th>&gt; 1:610000</th>
<th>Selected: 0</th>
<th>Snap</th>
<th>Replace</th>
<th>Zoom</th>
</tr>
</thead>
<tbody>
<tr>
<td>Projection</td>
<td>Location</td>
<td>Scale or Size</td>
<td>Selection</td>
<td>Snap</td>
<td>Selection Mode</td>
<td></td>
</tr>
</tbody>
</table>

- **Projection**: The name of the projection in use.
- **Location**: The current location of the mouse cursor, usually in longitude and latitude. Can be in X, Y native coordinates for the projection.
- **Scale or Size**: Show the current scale using either 1:xxx scale notation or by showing the width of the window in miles or kilometers. Enabled only if the component is projected (other than Latitude / Longitude).
- **Selection**: The number of objects selected, if any. Enabled for drawings.
- **Snap**: Shows "Snap" if a snap mode is active.
- **Selection Mode**: Current selection mode: Replace, Add, Subtract, Invert or Intersect.
- **Command**: Name of the current command.
**System Activity**  Changes shape into a 3D "bump" whenever Manifold is busy executing a command.

The location readout may be changed in **Tools - Options** to show the location in different ways. Likewise, a **Tools - Options** setting allows switching between 1:xxx scale mode or xxx mi size mode.

The illustration above shows alternate choices for location and scale readouts. The location shows the longitude and latitude in degrees, minutes and seconds format. The scale readout shows the width of the current view in kilometers.
Security and Access Control

Manifold System is designed to utilize Windows security and access control methods. The correct, most reliable path to security is to learn to use the security features in your operating system. The best way to assure fast, easy and reliable access control is to learn to use the access control features in your version of Windows.

Security and access control when doing GIS in Manifold revolves around sensible structuring of your Windows file system, organization of files and control over various resources such as network access to files, read/write permissions and more. Real security in Windows requires Windows Vista, 2003, XP or 2000 running on an NTFS file system.

For best management of security and access control, manifold.net recommends either Windows Vista, 2003, XP or Windows 2000 within a native Windows domain integrated with Active Directory. A Windows domain with Active Directory can operate with well over a million users. Informed use of Windows and features like Active Directory or Policies will provide enterprise-class security and access control capabilities.

Windows 2000 / XP / 2003 / Vista Resources

- Windows domains integrated with DNS - Domains in a Windows network are large and secure.
- Single or Multiple Domains - Enterprises can organize using a single, large domain or multiple domains with trust relationships between the multiple domains.
- Subdividing Domains into Organizational Units - Individual domains can be easily subdivided into organizational units. Control over those local organizational units can be delegated to local administrators using a simple wizard.
- Active Directory QoS Bandwidth Control - Allocate priority bandwidth to given users based on Quality of Service (QoS) of the network link, who they are, time service required, etc.
- User and Machine Groups - Both users and machines can belong to groups within Windows.
- Groups within Groups - A simple way of organizing access and other permissions is to put groups inside of other groups.
- Organizational Units - OU’s are used to create users with different account policies, to give control of certain user and/or machine accounts to a set of users, and to control user desktops through group policy objects.
- Sites - Windows sites can help an enterprise get organized geographically to improve network performance by compressing replication information that must travel through slower or costly WAN links.
- Distributed File System - A handy replacement for NT’s directory replication service.
- Encrypting File System - Keep files encrypted.
- Sophisticated Security Settings - Windows provides a very wide range of security options that may be applied to fit any organizational or operational need.
- Integrated support for Smart Cards - Smart cards authenticate users by providing a personal security certificate when inserted into the smart card reader. Windows now includes support for numerous smart card readers and media.

For information on configuring these and other security and administration features in Windows please consult any of the many books published on Windows Vista, Windows Server 2003, Windows XP and Windows 2000.
Where to Get Maps

Manifold reads so many different formats it is usually easy to find many sources for maps and their components: drawings, images, surfaces and geocoded tables or other data sets that may be merged into tables. In this topic we refer to data that may be imported as any Manifold component using the generic term "GIS data."

Manifold users working with the United States have a seeming infinity of free data sources to work with. Federal law in the United States places essentially all GIS data sets created by the Federal government into the public domain. Since Federal agencies in the US are the most prolific creators of GIS data in the world, US users are blessed with an abundance of data, usually free of charge. The usual problem in the US is choosing which one of many possible drawings or images is the right one to use.

The situation in the rest of the world is much different and for most users considerably more difficult than in the US. Except for the US, most governments do not place GIS data in the public domain. Most governments charge high prices for any GIS data for use outside of government. Some governments discourage or openly outlaw any competition to state cartographic monopolies. For these reasons, free GIS data for much of the world is difficult to obtain unless you are part of some government entity (military, public health organization, etc.) and can obtain free data.

General Sources for GIS Data

Manifold can read numerous formats for drawings, images, surfaces and tables. Sources for such data include:

The Manifold Site
A "starter" collection of base maps, including Manifold's base World map are available from downloads pages on the manifold.net web site. See the links to various downloads pages given in the Product Downloads page.

Note that some .map files available for download contain numerous drawings. Fetch drawings from these .map files using File - Import - Component to import just the drawing you want into your project.

All .map files on the downloads pages were all saved using the Compress .map files to save space option in Tools - Options to allow faster download. When opened for use the resulting projects may be considerably larger than might be expected from the size of the .map file on disk.

The Internet
The Internet provides more access to GIS data at zero cost than any other method. Become expert at using a good search engine and crawling the web for sources. When downloading from the web, be alert for any accompanying documentation that describes necessary projection information or other metadata. Acquire a fast Internet connection and a large hard disk as well. A revolution in fast and easy access to GIS displays comes from the use of Image Servers within Manifold.

Google KML / KMZ sites
A great way to get point data is to visit sites where Google Earth users exchange data as KML / KMZ files full of placemarks, such as the user forums at http://bbs.keyhole.com/ubb/. Manifold imports KML / KMZ format, so any file like that can be imported into Manifold as a drawing of points. Want a drawing showing all castles in Italy? Locations of famous battles? It's all there.

Government Sites
Agencies like www.usgs.gov, www.nima.mil, and www.census.gov (USGS, NIMA and the Census Bureau) on the federal level and various state and local agencies are placing more and more GIS data on the web for free download. Use government sites to get highly detailed drawings and maps for the area of interest. A very useful collection is at the National Atlas website at http://nationalatlas.gov/atlasftp.html

Non-Profit Organizations
Non-profit organizations such as www.ciesin.org often provide sites with GIS data that originated in government but which have not yet been made available by
Introduction

Universities Numerous universities operate web sites with free GIS data.

GIS Vendor Sites Many GIS vendors operate sites that provide highly detailed GIS data in the format of their GIS systems. ESRI, for example, provides free downloads of highly detailed international data in .shp format from their www.esri.com site. Manifold can read such formats. Read the vendor’s terms and conditions to see if your intended use of the data is permitted. Many vendors allow remarkably free usage of the free data provided on their sites so long as you credit them as the source of your data in publications.

Other Sites “Ezines” focusing on GIS as well as sites run by individual businesses or GIS enthusiasts provide hundreds of gigabytes of GIS data for free. Use a good search engine or ask about in newsgroups to find these.

Peer to Peer Exchange Internet forums like the GeoReference forum at http://forum.manifold.net, are a good place to meet other GIS users for exchange of GIS data. See the Help - Manifold on the Web topic for fast access to the Manifold online community and other resources.

Libraries Don’t overlook public libraries as a source of GIS data. These are usually best for locating paper maps or printed photographs for scanning, a less preferred route to GIS data than direct electronic download via the web. Larger libraries will often have CDs on hand from the Census Bureau or other agencies. Bring a laptop into the library and copy the data onto your hard disk for later use. If it’s public data, it’s your right to copy it.

Government Libraries Many government agencies have libraries that are open to the public. USGS has major libraries in Virginia, Colorado and Northern California that provide access to hundreds of CDs of USGS-published GIS data. Bring your laptop.

Federal Depository Libraries By Federal Law, agencies are required to provide copies of their publications, including electronic publications like GIS data CDs, to Federal Depository Libraries. University libraries will often function as a Federal Depository Library for their particular location. If a library participates in the Federal Depository system, you must be granted free access to that library and allowed to use the Federal materials on file. Again, bring your laptop. Search the web for information on the nearest Federal Depository library to you.

University Libraries University libraries are often open to the public for free or for a nominal fee. Any good university library will have a wealth of GIS data on CD available for use. Most have no problem with use of personal laptops or copying of GIS data that is in the public domain.

Government Sales USGS provides numerous GIS data titles on CD for low costs. The Bureau of Transportation Statistics (BTS) will ship CDs to you for free. Numerous other agencies will sell data. Most government data in the US is sold for reasonable prices, but a few agencies charge illegally high prices. If confronted with a demand to pay $1500 for a couple of CDs, file a Freedom of Information Act request instead. Agencies cannot charge more than the cost of duplicating the data.

GIS Data Vendors For those parts of the world not covered by publicly available NIMA data sets, the only options may be to create your own data or purchase it from a commercial vendor. In the US, the only sources of very timely and high accuracy data are commercial vendors. Vendors like TeleAtlas (www.teleatlas.com) provide good
coverage of Europe. Manifold can read all common commercial formats.

**Satellite Photography Vendors**

Commercial satellite photography vendors such as IKONOS ([www.spaceimaging.com](http://www.spaceimaging.com)) and SPOT ([www.spot.com](http://www.spot.com)) sell satellite images from their huge inventory of existing images. Some companies will even snap a custom picture for you from orbit of the location you desire. Other sources include archives of Landsat and other data.

**Freedom of Information Act**

The United States guarantees access to virtually all unclassified Federal data via the Freedom of Information Act (FOIA). If you see a map created by any Federal agency, you should be able to get the GIS data that was used to create the map with a simple FOIA request. Use a good web search engine to find resources and step-by-step instructions for using FOIA. Before filing a FOIA request, please check carefully to make sure the data you seek has not already been published.

**Key Internet GIS Data Sources**

Internet URLs change frequently. The URLs below were active when this documentation was created. If the URLs are changed, usually drilling down from the main web site for the agency or organization will find the current URL. In addition to USGS Geodata, the National Atlas, the Census Bureau and NIMA, there are many, many other resources on the web. For example, many state governments operate web sites from which GIS data may be downloaded.

**Image Servers**

Probably the greatest change ever in the GIS industry, introducing immediate access to the largest set of worldwide GIS data ever, has been the provision of Image Servers within Manifold. Read the image server topics and the tutorial to learn how to immediately generate map backgrounds and satellite photography for almost anywhere on Earth. The use of Open Street Maps (OSM) image servers in particular provides free, public domain mapping for virtually anywhere on Earth.

**USGS Geodata Downloads**

[www.usgs.gov](http://www.usgs.gov) is one of the best sites in the US for GIS data products. Numerous USGS servers provide many different sites from which data may be downloaded.

[http://edc.usgs.gov/doc/edchome/ndcdb/ndcdb.html](http://edc.usgs.gov/doc/edchome/ndcdb/ndcdb.html) - A Manifold favorite for USGS data. See this site for documentation and detailed information on the various data sets provided. This site provides free downloads for:

- **1:2M DLG** - 1:2,000,000-scale drawings showing boundaries, major roads, hydrography, etc. for the United States. Suitable for background maps and overviews at the state level.

- **1:100K DLG** - 1:100,000-scale drawings showing features in greater detail. Used by many for roads and similar features at the county or city level.

- **1:24K DLG** - Actually published using SDTS format, these are the vector drawing equivalent of USGS paper topological "Quad" maps. Very highly detailed roads, hydrography and much more. Used for local work.

- **1:250K DEM** - 1:250,000-scale Digital Elevation Modules (terrain elevation) suitable for use as surfaces at the county level.

- **1:24K DEM** - 1:24,000-scale Digital Elevation Modules published in SDTS format suitable for use as surfaces at the local level. Equivalent to "Quad" maps. Extraordinarily detailed. Use the SDTS drawing importer to import these (a surface will automatically be created even though a drawing importer was used). Note: USGS is now moving free downloads of SDTS DEMs to third parties.
LULC - In 1:250,000-scale and 1:100,000-scale versions. Land use and land cover data provides information on urban or built up land, agricultural land, rangeland, forest land, water, wetlands, barren land, tundra, and perennial snow or ice. Associated data sets display information in five data categories: (1) political units, (2) hydrologic units, (3) census county subdivisions, (4) Federal land ownership, and (5) State land ownership. The 1:250,000-scale resolution maps yield spectacular maps when thematically formatted in Manifold.

NCLD - A collection of raster data sets, one per state, that provides extensive land use data for the US as rasterized data sets. Derived from the early to mid-1990s Landsat Thematic Mapper satellite data, the National Land Cover Data (NCLD) is a 21-class land cover classification scheme applied consistently over the United States. The spatial resolution of the data is 30 meters and mapped in the Albers Conic Equal Area projection, NAD 83. These are imported into Manifold as surfaces using the Raw Binary Files importer. See the Import a Raw Binary File - NLCD topic for an example.


US Bureau of the Census http://www.census.gov - The Census Bureau has begun placing data on line for download. Drill down to subsequent pages, like http://tiger.census.gov/ to find specific data sets. There are many pages that just serve images: drill deeper to find the actual GIS data sets for free download. A favorite Manifold site is http://www.census.gov/geo/www/cob/ the home page for Cartographic Boundary Files.

TIGER/Line - A highly detailed, complex GIS data set providing streets with address ranges, boundaries and numerous other detailed features for the entire US. The basis for most commercial street maps. See the Census Bureau's Tiger pages.

Boundary Files - States, Counties, Congressional Districts, Metropolitan Areas, Urbanized Areas, Incorporated Places, County Subdivisions, School Districts, Voting Districts, Census Tracts, Census Block Groups and much more. Usually organized by State. Download in .e00 format so that they will import by default into Manifold with the correct NAD83 datum used by most files. Alternately, when importing .shp versions (preferred by many people because the result of importing .shp files are simpler, better organized projects) use Edit - Assign Projection to change the datum to North American 1983 (mean for Conus).

ZCTAs - ZIP Code Tabulation Areas are the Census Bureau's representation of ZIP codes as areas. We predict these will become a standard for demographic work.

Gazetteer - Places, ZIP and other centroids.
find maps and geographic data at the NIMA web site. Use the Freedom of Information Act to pry public data out of NIMA.

VMAP1 - VMAP Level 1 data: the world on 232 CDs at exquisite detail (1:250,000 to 1:100,000-scale). NIMA has released some data, but not all. File a Freedom of Information Act request for what you need if they do not have it available for download.

VMAP0 - VMAP Level 0 data. Updated edition of the famous Digital Chart of the World.

DTED - DTED Level 0, terrain elevation data for the world.

DOI - Georeferenced SPOT satellite imagery for much of the world.

JOGA - Joint Operations Graphic - Air, JOG-A (1:250,000 scale) digital raster graphics images. (scanned paper charts).

ONC - Operational Navigation Chart, ONC (1:1,000,000 scale) digital raster graphics images. (scanned paper charts).

TPC - Tactical Pilotage Chart, TPC (1:500,000 scale) digital raster graphics images. (scanned paper charts).

Bureau of Transportation Statistics

http://www.bts.gov - A "must have" CD for every serious GIS person is the National Transportation Atlas Database (NTAD). In addition to key transportation layers it has many general purpose GIS data sets as well. At the current writing, BTS will ship you an NTAD CD free. It may be "ordered" at zero cost from the BTS web site. Several of the data sets on the Manifold downloads site come from the NTAD. BTS also provides free streets for the entire US in the form of their "DynaMap" street series (vintage 2000).

CIESIN

http://www.ciesin.org - The Center for International Earth Science Information Network at Columbia University: see their Data Resources page for links to the archive of Census related products (detailed demographic data), fundamental GIS layers for China, world population data and more.

International GIS Data Sources

National Imagery and Mapping Agency

http://www.nima.mil - The primary source for public domain international data. See notes in the table above. Just how much data NIMA will continue to provide in a "high security" environment is still open to question.

Bureau of Transportation Statistics

http://www.bts.gov - Provides transportation GIS layers for Canada and Mexico as part of its North American Atlas Database (NORTAD) free CD.

Geobase

http://www.geobase.ca - A new Canadian site providing a wide array of Canadian data, including Administrative Boundaries, Canadian Digital Elevation Data (CDED) Canadian Geodetic Network, Geographical Names of Canada, Landsat-7 images and the Canadian National Road Network (NRN).

Geogratings

http://geogratis.cgdi.gc.ca - Cool Canadian data from Natural Resources Canada for free download, plus links to other free data sites. Includes the National Atlas of Canada.

NASA

NASA is beginning to put large amounts of data online for free download. For example, virtually complete Landsat imagery for the entire Earth in MrSID format (easily
converted into Manifold images) is at:

https://zulu.ssc.nasa.gov/mrsid/

NOAA GLOBE

The Global Land One-km Base Elevation (GLOBE) project provides the latest, most modern, public domain terrain elevation database for the entire world. Currently at http://www.ngdc.noaa.gov/seg/topo/globeget.shtml, the data set may be downloaded in 16 "tiles," most of which are very large files (over 100MB each) when unzipped. Import using the DEM GLOBE Files (*.*) setting in File - Import - Surface.


State Cartographic Bureaucracies

When all else fails and you cannot find your data for free or by trading with colleagues online, the last resort is to purchase it from the local cartographic bureaucracy. For example, the Ordnance Survey (OS) in the UK is the authorized state cartographic monopoly that sells data for the UK. The data is quite good, but because the OS is such an ineptly run bureaucracy it will cost you perhaps 100 times more than an efficiently run organization would charge.

Other resources

If you can find it online in one of the common GIS interchange formats (.shp, .mid/.mif, .e00) it's almost certain you can read it in Manifold. Many vendors provide GIS data for free without restrictions.

Commercial Sources

Although the emphasis on the above is on free data sources, there are many fine companies providing GIS data for a fee. Such companies often begin with government data for a particular region and then enhance it by reorganizing the data into more user-friendly form, correcting mistakes, updating the data and so on.

Creating Your Own Data

Always keep in mind the possibility of creating your own data sets. This may be required for some international users if commercial or government data is not available. There are several methods to create a custom map:

Acquire GPS data points

Manifold includes a connection to GPS devices controlled by the GPS Console. For small regions such as ranches and farms it's remarkably easy to create a map by driving or walking boundaries with a GPS device connected to a laptop running Manifold. GPS data points can be used to mark control points used to georegister aerial photographs.

Scan a paper map

Although almost all modern maps are created digitally, many maps still exist in paper form. This is especially true of historical maps. One can scan a paper map as an image and then georegister it using control points that appear in digital maps that map be obtained. Scanned maps can be used as images (slow and inefficient) or converted into drawings using tracing.

Scan a printed photograph

Printed photographs may be scanned and georegistered like scanned paper maps. Many libraries have archives of aerial photographs that can be scanned. Modern flatbed scanners are very light and are easy to carry into a library together with a notebook computer for scanning.
of materials that are not allowed to circulate outside the library.

**Image Servers**

Overhead satellite photography for much of the world is now available in very high resolution at no charge for many applications using a new generation of image servers like TerraServer, those used by NASA WorldWind, Yahoo!, Google Earth and Virtual Earth. The Manifold image server interface enables fast and easy use of many image server modules within Manifold. Users can easily create their own digital maps of roads and other features by rapidly clicking over an image server background.

**Custom aerial photography**

Custom aerial photography ranges from inexpensive to very expensive custom work. Most GIS purposes can be accomplished with inexpensive, ad hoc aerial photography. Visit a local airport near the area of interest and hire a pilot and a Cessna for an hour. Shoot a roll of film (or use a quality digital camera) from a suitable altitude on a clear day and many images will be very useful once scanned. Georegister them using control points. **Tip:** Crack open the door a bit and shoot down through the crack so no glass is between the camera and the scene. Search the web for other aerial photography tips.

**Custom satellite photography**

Stock satellite photographs with two-meter resolution will frequently sell for under $100. Custom satellite photography from commercial operators is occasionally available for under $500 an image if the satellite viewing swath traverses the target area. Prices will no doubt come down as the amount of stock photography and satellite competition increases.

**Tables as Sources of GIS Data**

Much data is published as tables in various forms that can be used in GIS work even though the tables are not geocoded. A good example is the **Counties** series of data sets published by the Census Bureau on CDs with titles like **Counties 1999** or the **City and County Databook**. Such data sets are collections of tables, often in .dbf dBase II format.

A typical table might consist of records of demographic data for each county where each record has the county **FIPS** (Federal Information Processing Standard) numeric code identifying the county together with other data fields. We can use such tables in Manifold if we first import a drawing that shows counties that also has a **FIPS** code field for each county. We can then form a relation between the drawing’s table and whatever table was imported from the demographic data set. This same technique can be applied to use tabular data in many different types of GIS work.

Tables may also appear in electronic documents using non-database formats. For example, one might encounter a PDF file containing a table, an HTML web site page showing a table or a Microsoft Word .doc file that contains a table. The experienced user will learn Windows techniques for transforming such data into a format that can be read as a table by Manifold.

For example, we might highlight some tabular text in a PDF file, paste it into a Microsoft Word document and the use Word to convert the text into a real table. We could then “clean it up” if need be in Word and save it either as an HTML document containing a single table or write it out as a .csv (“Comma Separated Values”) text file. Sometimes clever use of search and replace to convert tabs to commas or other character manipulations within Word will help us create the table we desire for import into Manifold.

**Notes**

Respect copyright laws: It is important to become familiar with the copyright laws that apply in your jurisdiction. Knowing the law will help you avoid accidentally ripping someone off, and it will also prevent unscrupulous data providers from ripping you off as well.

Web sites change constantly. The URL’s mentioned above or the availability of the data sets described may have changed since the publication of this document.
A Call to Arms

When living in a free society it is easy to become complacent about one's freedoms and rights. If you live in the United States you have a right to get and to use public data, but that right will disappear if you are not vigilant about maintaining it. Even as Internet makes it easier and cheaper than ever for agencies to provide public access to public data, some agencies are using Internet to make it more difficult to get access to public data.

A recent trend, for example, is for many state and federal web sites to present only PDF files or web-served images of data instead of the real data themselves in some publicly accessible GIS format such as .shp. Even in cases where the law requires delivery of public data in an open, non-proprietary format, some agencies will deliver data only in a proprietary format, such as "MrSID".sid format, that cannot be accessed through genuinely public means. This game can be played in a very devious manner.

For example, a common misconception is that .sid format is OK to use for public data because the makers of the format at Lizardtech (an appropriately reptilian name) provide a "free viewer" which can display the data and export it to open formats such as GeoTIFF. However, if you try to acquire and to use that "free viewer" you'll find that it is a licensed, proprietary tool which requires users to agree with Lizardtech's grossly objectionable legal positions as a legal requirement of using the tool. It is a bit like justifying illegal restrictions on your right to vote by saying that they are not a factor if you sign an agreement saying that you can vote, but only on the condition that you agree your voting rights can be taken away at will.

Whenever you run across a web site that provides access to only images of the data or which uses proprietary formats, take a moment to email the people affiliated with that web site to ask them how you can get access to the real data or to data in non-proprietary form. In the case of proprietary formats, explain to them the data is not in non-proprietary form if you must agree to objectionable, proprietary legal requirements to view it or to translate it into open form. If the real data is not available, encourage them to post links that allow download of the actual data. If they refuse, ask them to whom in their agency a Freedom of Information Act (FOIA) request should be addressed to get the data. Most states have a statute equivalent to the FOIA that allows you to get access to public data whether the agency likes it or not.

Some agencies mean well: they know that GIS usually costs many thousands of dollars per license so they put effort into providing images in the belief that the public cannot afford to work with the real data. Bring them up to date: tell them that Manifold makes it possible to work with the most sophisticated data to as great a degree of sophistication as one chooses at prices no higher than ordinary Microsoft Office applications. The more sophisticated the data is, the more important it is that users be able to work with it as real data.

See the Public Access to Public Data essay for a rant in support of public access to government data.
Performance Tips

Manifold includes many capabilities previously found only on software that costs tens of thousands of dollars per license. That the software includes such sophisticated capabilities does not imply that any personal computer, no matter how limited or slow, will be able to adequately support use of such capabilities. In particular, work with large images will require lots of RAM and a fast processor. The thousands of dollars we save by purchasing Manifold can be invested into more RAM and more powerful processors that can serve us in other applications as well.

Most Manifold projects using vector drawings on modern computers will operate so fast that thinking about performance is not necessary. When working with large projects or large images, though, we may want to squeeze every bit of performance out of our systems. The following tips will help you achieve the greatest possible performance on your desktop when working with large maps. Many of these suggestions are generic suggestions that will help you create and operate the fastest Windows system possible.

Important Configuration Notice

Very Important: Before starting Manifold, go to the Windows Control Panel Display dialog Effects tab and make sure that the Show window contents while dragging check box is not checked.

- In Windows XP this option is found in the Control Panel's Appearances and Themes - Display choice under the Appearance tab by pressing the Effects button.
- In Windows XP or Windows 2003, from the Start button open the Control panel and then open the Display dialog. Click on the Appearances tab and then press the Effects button. Uncheck the Show window contents while dragging check box.
- In Windows 2000, Windows ME and Windows 98 open the Control panel and then open the Display dialog. Click on the Effects tab, and uncheck the Show window contents while dragging check box.
- In Windows Vista, open the Control Panel and click Appearance and Personalization. Click Customize colors, then click Open classic appearance properties for more color options and click the Effects button. Uncheck the Show window contents while dragging check box.

Checking this box will greatly slow down the system when displaying complex maps and images because it forces a refresh of the window contents with each minor change in mouse position while dragging the window. Make sure the box is unchecked.

Hardware

- Install lots of RAM memory, enough to maintain your entire project in RAM memory if possible. It does no good to have a fast processor if the fast processor is waiting around for virtual memory to be paged back in from hard disk. Don't forget to include RAM required by Windows and background processes. Memory is very inexpensive, so install as many gigabytes as you can. On newer systems, install as much as you can afford, using four gigabytes if you plan on working with large projects or multiple Manifold instances. RAM has become very inexpensive: a few hundred dollars can buy massive amounts of RAM, multiple gigabytes.
- Since 64-bit Manifold can make use of far larger amounts of RAM than 32-bit Manifold, it makes all the more sense to install lots of RAM (3 GB, 4 GB or more if possible) and / or increase the size of the Windows pagefile.
- Install a fast processor. If your budget allows a choice between substantially more RAM or a slightly faster processor, get more RAM and a slightly slower processor. If you are working with very large images and surfaces that will not fit into RAM, then it is more important to install a significantly faster processor and a really fast disk drive.
- Install a 64-bit Intel or AMD processor, preferably a quad core or dual core processor. Modern multi core processors run very, very fast and will offload some tasks at a price that is barely more than charged for single-core 32 bit processors. They are remarkably inexpensive for the speed and quality they give. Shop around and often you'll find motherboard and very fast multi core, 64-bit processor combos for $300 or less.
- Use multiprocessor machines or multi-core processors. Each new release of Manifold adds more functions that are multi-threaded. For example, Manifold will use more than one thread to render image libraries if more than one processor or processor core is available on the computer system. Therefore, image libraries will render faster on multiprocessor or on multi-core processor systems such as those using a quad-core or dual-core Intel or AMD processor.
- Use a modern motherboard with performance features. For example as of this writing some motherboards support full-bandwidth SLI for multiple graphics cards, RAID SATA and can accept 8 GB of RAM. Nice! Extremists may want to try out the over-clocking features available with modern
motherboards, cool their CPUs with water-cooling and other tricks. See the gaming sites for tips and tricks on maxing out motherboard performance.

- Install an NVIDIA GPU card supported by NVIDIA CUDA. Manifold can use CUDA to dramatically accelerate certain tasks. CUDA-enabled NVIDIA GPUs are so ubiquitous that no matter what your budget you can probably get a CUDA-enabled system for no additional cost. Whatever system you buy will have graphics capability in any event, so you may as well procure NVIDIA-based capability that can also dramatically increase performance through CUDA.

- Install a very large disk drive. Large disk drives are faster than smaller drives, because the average access time given is calculated over a larger capacity. It's better to install a larger, reasonably fast drive than a smaller drive that promises extra-fast transfer. Better still, get enough memory to never have to hit disk. An additional bonus to a large disk is that if you have 300 gigabytes of free space you will not hesitate to save interim versions of a project. Frequently saving projects is cheap insurance against undoable user errors and other time wasters.

- Get a fast hard disk, spinning at 7200 RPM or faster and using ATA100 or more recent interface such as SATA. This is especially important if you will work with large images or drawings, which will involve a lot of disk accesses.

- Install a second or third hard disk, so you can keep your Windows page file on a separate disk volume. If your entire job does not fit into available RAM Windows will have to swap it out to disk. Swapping often occurs at the same time that other disk accesses need to be made. With two disk drives in play Windows can launch the heads on one disk drive for read/write operations for swapping at the same time that the heads on the other drive are moving into position for other read/write operations, thus reducing the average access time. This is less important if you have lots of RAM, but as inexpensive as RAM has become disks can be even cheaper: it can be very inexpensive (under $50) to add a second, small, fast hard disk to host the Windows page file. An ideal situation is to have three disk drives, one for the page file, one for TEMP files and one for ordinary working files.

- Think twice before buying SCSI. It's often faster to use the latest generation SATA drives and to spend the money saved over SCSI on larger drives and more drives (for independent action when using page files and temp files). Keep your system page file, TEMP folder and user profile directories on a drive separate from your working files.

- Consider installing multiple disks in a striped RAID configuration for maximum disk throughput. For very large images and surfaces this might not be such an expensive option as it sounds, since large, fast hard disks can now be had for $100 each or less. Whether or not it is faster to use striped RAID than it is to use multiple hard disks for different Windows folders and page files is often a toss-up. Most technical users at manifold.net use striped RAID arrays when they want maximum speed because that requires less thought.

- For 3D terrain windows get a fast graphics card with OpenGL support in hardware and lots of local graphics memory. These are now very cheap, with supercomputer-class GPUs and 256 MB of RAM being sold in graphics cards for as little as $100. It is critically important to use good drivers. Quite often the fastest drivers for your version of Windows will be supplied by the chip vendor who makes the graphics chip in your graphics card. This can make the difference between smooth “fly through” motion and very jerky, a-few-seconds-per-frame motion in terrain views. Manifold requires a functioning OpenGL subsystem to display terrains. If there are no OpenGL capabilities in the system terrain windows will be blank when opened. See the discussion in the Terrains topic.

- For maximum terrain viewing performance, use SLI-capable nVidia PCI Express graphics cards in an SLI-capable motherboard to team multiple graphics cards for rendering. Prices on graphics cards are dropping rapidly: as of this writing, installing two high-end, SLI-capable nVidia graphics cards with 256 MB of RAM each costs a total of $250, an amazing deal for the resultant throughput. It is often faster to use two cards via SLI than it is to spend disproportionately more money for a single card that uses the very latest, super-hot graphics chip. For example, two SLI cards using slightly downrev, but still awesome chips might be had for $150 each and the combination could end up being as fast as or faster than the latest superchip board at $750 each. Of course, if money is no object, get two of the latest boards!

- Use a large video display with high resolution. Larger, high-resolution displays will allow you to keep more panes, toolbars and windows open at the same time. Work goes faster when all controls are in sight. If you are stuck with a small screen or low resolution, learn to use the ALT-SHIFT keyboard shortcuts to rapidly open and close panes. See the Windows topic for tips on keyboard shortcuts.

- Get a flat panel LCD display that accepts DVI direct digital video input and install a graphics card with DVI output. The direct digital connection will drive the LCD flat panel at superb clarity and crispness to reduce eyestrain and fatigue, allowing you to work longer and more effectively.

- Use a wheel mouse with additional buttons like the Microsoft Intellimouse. The additional controls can be very useful for navigating within windows.

**Operating System and Other Software**

- **Very important:** Go to the Windows Control Panel Display dialog Effects tab and make sure that the **Show window contents while dragging** check box is not checked. Checking this box will greatly slow down the system when displaying complex maps and images.
Use any version of Windows 7, Windows Server 2008, Windows Server 2003 or Windows XP. More recent Windows editions provide much better utilization of large memory than do Windows Me or Windows 98. Vista is great, truly superb, but it is not at the present writing as fast as XP.

Use a 64-bit processor and install an x64 Windows version such as Windows 7 x64, Windows Server 2008 x64, Windows Vista x64, Windows XP x64 or Windows Server 2003 x64. Run 64-bit Manifold x64 editions.

Install the latest updates and service packs for the software you are using. Windows, DBMS vendors and others spend billions of dollars on improving the quality and performance of their software by issuing updates, often free updates. Take advantage of that.

Use NTFS for the file system. Do not use FAT or FAT32 (FAT file systems have serious drawbacks in terms of security, safety and performance).

Avoid creating many partitions. Ideally, set up one partition for the entire drive C:.

Set up a fixed-size Windows page file that is far larger than ever will be necessary, several gigabytes or larger in the case of projects involving very large images. This is faster than a dynamically re-sized page file. In Windows 2000 open System in the Control Panel. In the Advanced tab click Performance Options and under Virtual memory click Change. Set the Initial size the same as the Maximum size. In Windows XP, open System in the Control Panel, then in the Advanced tab’s Performance section click Settings. In the resulting Performance Options dialog’s Advanced tab’s Virtual Memory section click Change and then set the Initial size the same as the Maximum size. In both cases the size should be very large, over 2000 MB if possible.

Allow a large enough page file for multiple instances of Manifold if you will be launching more than one instance of Manifold and copying and pasting between them. More than 2.5 gigabytes per Manifold instance is not necessary.

Install the latest version of Internet Explorer. IE installs fresh versions of Windows system facilities such as VBScript and JScript scripting engines, the XML parser and other facilities that are used by Manifold. As newer versions of IE install faster and better facilities, your Manifold installation will also become faster and better.

Avoid running other applications in background.

Minimize the use of memory-resident services. Keep only absolutely vital services on the traybar. The only program sitting on traybars at manifold.net is a readout for system time. A few systems have a volume control icon and some systems have an icon indicating the status of a local SQL Server or Oracle DBMS server. Compare that to the collection of seven or eight icons often seen in traybars when many intrusive, memory-resident consumer applications have been installed.

**Very important:** Do not operate a virus checker in real-time scanning mode. Instead, schedule a regular weekly or nightly virus check. Real-time virus scanning will have a serious impact on system performance because the virus scanner will analyze all of the many temp files created by Manifold in the normal course of work. This tip is especially important when working with large images or surfaces.

When running WinAmp, Windows Media Player or other audio players at the same time as Manifold, switch off visualization modes to reduce processing overhead. Switch off the equalizer or effects as well if you need every processor cycle. If you do not have enough RAM to avoid frequent paging to disk, run your audio playlist over your local network using a different machine as a disk server to avoid tying up the local disk.

If you have multiple hard disks, keep your TEMP directory and page file on one disk and your applications and .map files on a different disk. When working with very large images and surfaces it helps to keep the TEMP directory and system page file on physically separate hard disks. This allows the disk drive heads to seek independently from each other as various files are used. This is less of a factor if you have so much RAM that disk accesses are minimized, and more of a factor when working with large projects that don’t fit into RAM.

When working with server-based OLE DB providers such as SQL Server or Oracle, users are strongly encouraged to maintain primary keys in all tables linked into the Manifold project. A side effect of how such servers interact through OLE DB is that if the table does not have a primary key, performance will be greatly reduced.

Use ADO .NET to connect to DBMS servers instead of ODBC or OLE DB. Use the native Oracle Call Interface (OCI) when connecting to Oracle.

If you have installed a CUDA-capable NVIDIA graphics engine, make sure you also install the CUDA libraries so that Manifold can take advantage of the graphics engine to accelerate computation. CUDA libraries are a built in part of NVIDIA-supplied GeForce drivers, but might not be a part of default Windows drivers. Install the latest drivers from NVIDIA to be sure.

Learn how to operate and administer Windows proficiently. Use a good Internet search engine (such as google.com) to find web sites that teach how to tune your Windows system for maximum performance. If your Windows system is tuned for faster operation then applications that run within Windows, such as Manifold System, will also be able to run faster. Simple Windows maintenance such as defragmenting hard disk storage can result in noticeable improvement in Windows performance.

**Manifold System**
Run 64-bit x64 editions of Manifold System within 64-bit x64 Windows.

Turn on the **Render data progressively** option in the Tools - Options dialog (on by default). This allows the system to be responsive while large drawings, themes, images and surfaces are rendered in stages. Quite often we are rendering a drawing simply to orient ourselves with the attention of zooming in to some smaller region, so being able to begin the zoom command before the drawing has finished rendering is a big time saver. When a component is small enough to render very quickly, progressive rendering will not be noticed because the component will be rendered in a single pass. **Note:** Lengthy operations such as some transforms automatically suspend progressive rendering and resume it upon completion, so that the system does not lose time rendering what might be obsolete data.

Re-project drawings, images, and surfaces into the same projection specified in maps that use these drawings, images and surfaces. All projection parameters must be the same for this to help and not just the name of the projection in use. This is critically important when working with very large images and surfaces. If this is not done then everything from displaying maps in their own window or displaying a layout that includes the map can end up being very slow.

If a map will consist of a large image (or surface) and several drawings in a different projection, create the map using the image and then add the drawing layers. The map will then use the same projection as the image. Large images and surfaces are much slower to re-project on the fly than are drawings, so it is faster for the map to use the same projection and to re-project the drawings for display than the other way around. When working with large images, it is critically important to use the projection of the image for any maps that contain that image.

Turn off any layers you don't need to see in a map.

Close any unnecessary map, drawing, or image windows. Every window that is open will need to be redisplayed on any changes.

When doing analytic work that does not need much visual interaction, use smaller map windows and zoom far into the map so relatively few objects are on screen. The fewer objects that are visible, the fewer need to be computed for rendering and redrawn.

**Uncheck View - Refresh Auto** so no refreshes occur unless you command them with **Refresh**.

Don't use high-resolution data when you will zoom out to levels where the details blur together. Resample the data with Normalize Topology (for drawings) using a lower Location Precision value to create a lower resolution equivalent that makes sense at the zoom level you need. For example, don't use high-resolution shoreline data if the map will show an entire world at once. The details will not be visible on screen, but Manifold will still need to re-compute the display based on a huge level of detail. Working with lower resolution data will also make many commands, such as Dissolve, operate much faster. Many drawings, such as those destined for thematic presentation, can be made ten times less complex/smaller without any objectionable visual effects.

Don't use raster images when drawings are more accurate, faster, and provide more information content. See the Images can be Inefficient essay.

When working with images stored in "lossy" formats such as .jpeg, resample (resize) the image down to the level of resolution truly kept with the .jpeg compression. Expanding the compressed image to a large size with many pixels in X and Y is a mirage, because the detail for high resolution has already been lost. You may as well enjoy the faster speed obtained by using an image at pixel resolution commensurate with the true image information it actually contains.

If you only need to work with part of a large data set, take a moment to cut that part out as a separate drawing or image or terrain and then work with only the necessary part.

**Uncheck the Preview box in image commands with large images if you don't need to see a preview.**

RGBa images are larger and take more time to display than ordinary RGB images. Don't use RGBa images unless you need to use RGBa pixel transparency, or if ordinary invisible pixels within an RGB image can do the job.

Use Zoom Ranges to present the level of detail required - Suppose we would like to show a national map for user orientation when zoomed out but we would also like to show very detailed shorelines when users zoom in. Create a map using two drawings: a highly detailed drawing plus a second, low resolution drawing created from the first by generalizing with Normalize Topology to a lower Location Precision value. Assign zoom ranges to the drawings so that the low-resolution drawing is the only drawing displayed when zoomed out and the high-resolution drawing is the only drawing displayed when zoomed in. When users browse the map in a zoomed out view the display will be fast because a fewer number of coordinates need be used to display the low-resolution drawing. When users are zoomed in the high-resolution drawing will become visible; however, only a portion of the high-resolution drawing will be seen in a zoomed in view so the display will still be fast.

Store the .map file used on a fast, local hard disk - When Manifold operates it accesses the .map file and any temp files created. If the .map file is located on a slow hard disk or on a different machine that must be accessed via a local area network then performance will not be as fast as if the .map file was immediately available on a local, fast hard disk. If you have a very fast network and fast servers this effect might not be significant.

When importing drawings, don't import data fields that will never be used. Many data fields will slow the system down. If a drawing has been imported with superfluous data fields, edit tables using Design to eliminate the unnecessary fields. When working with commands such as Dissolve, check the Transfer box in image commands with large images if you don't need to see a preview. RGBa images are larger and take more time to display than ordinary RGB images. Don't use RGBa images unless you need to use RGBa pixel transparency, or if ordinary invisible pixels within an RGB image can do the job.
Rules for the tables being used to make sure that no fields are being transferred that are not necessary to transfer.

- Uncheck the **Compress .map files to save space** option in Tools - Options to eliminate .map file compression. This will result in larger .map files on hard disk, but saving .map files and opening .map files usually will be a much faster process. Oddly enough, if a project includes very large images or surfaces and you have a very fast processor, it may be faster to use compression because the time to fetch and decompress a smaller sized file may be shorter than the time required to fetch a larger, uncompressed file.

- Always acquire and install the latest edition of Manifold System or the latest Service Pack. Each new build of Manifold System includes optimizations and other improvements. Some optimizations will result in dramatically faster performance in the functions that have been optimized.

- When working with large images and surfaces, keep the image or surface stored in a Manifold .map file. When a large image or surface is opened for the first time after it is imported Manifold will build a series of intermediate views that are used for faster panning and zooming at less than full resolution. Building these intermediate views takes time, so the first time a large image or surface is displayed the window will open very slowly. After that, viewing will be fast. If the image or surface is stored in a .map file, the intermediate views will be stored in the .map file as well (as a built-in part of the saved image or surface). Opening an image or surface already saved in a .map is therefore fast even the very first time the image or surface is opened.

- When working with large images, if they need only to be viewed in a read-only way, consider storing them as compressed images so they load and display nearly instantly.

- Store large images within a fast DBMS server such as DB2, MySQL, Oracle, PostgreSQL or SQL Server.

- Complex queries take longer to accomplish than simple queries. A SELECT clause with some simple computations in the WHERE clause will scale linearly; that is, handling 100,000 records will take about 10 times longer than handling 10,000 records. A SELECT with a GROUP BY or ORDER BY clause will scale slightly worse, perhaps using n log n scaling where handling 100,000 records may take, for example, 12 times longer than handling 10,000 records. Joined SELECTs with spatial computations may scale even worse.

- When re-projecting using the **Clip coordinates** option, make sure to read and take heed of the notes at the end of the Orthographic topic.

- Use native tables instead of linking tables. Tables within the Manifold project are the fastest, shared tables (with caching turned on) from an Enterprise server will be slightly slower and linked tables from an external DBMS provider are the slowest. Retrieving data from a table linked to a fast external data source will take at least 10 times as long as retrieving the same data from a table inside the Manifold project.

- DBMS software other than SQL Server may not always favor OLE DB over ODBC. Some databases have slow OLE DB drivers and fast ODBC drivers, and other databases (like SQL Server) have fast OLE DB drivers and slow ODBC drivers. The performance of an ADO.NET provider might or might not be better than OLE DB or ODBC drivers, depending on whether the provider is entirely implemented in managed code or not and on a multitude of other factors.

- When using external DBMS tables, use the fastest connection possible, which may vary depending on the DBMS in use. For example, when connecting to Microsoft's SQL Server, ADO.NET is faster than OLE DB and OLE DB is faster than ODBC. ADO.NET is **much** faster than ODBC. Consider that connecting to a remote SQL Server can take much longer than connecting to a local SQL Server. For example, suppose that connecting to a table inside the Manifold project takes 1 unit of time. A very rough guide to equivalent times to access the same table using different connection and DBMS technologies is listed in the accompanying table. Clearly, it is very unwise to connect to remote SQL Server installations using ODBC. **Important**: Tables linked from ADO.NET data sources are always read-only. If read-only access is acceptable, ADO.NET is often the fastest possible connection and should be used in preference to ODBC or OLE DB for that reason.

<table>
<thead>
<tr>
<th>Time</th>
<th>Source for Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Native table (inside the project)</td>
</tr>
<tr>
<td>8</td>
<td>Linked from a local MDB file</td>
</tr>
<tr>
<td>10</td>
<td>Linked from a local SQL Server via ADO.NET</td>
</tr>
<tr>
<td>40</td>
<td>Linked from a local SQL Server via OLE DB</td>
</tr>
<tr>
<td>6000</td>
<td>Linked from a remote SQL Server via OLE DB</td>
</tr>
<tr>
<td>250</td>
<td>Linked from a remote SQL Server via ADO.NET</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Time</th>
<th>Source for Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>6000</td>
<td>Linked from a remote SQL Server via ODBC</td>
</tr>
</tbody>
</table>
Introduction

- There is no performance difference between storing geometry in external tables (linked drawings) or internal geometry, once external geometry is refreshed. Refreshing geometry from an external table will likely take as much time as it takes to run a fairly modest SELECT query, with the external data source being the bottleneck.
- When querying external tables, running a Manifold query analyzes the entire table and fetches each record from the server, so it might sometimes be faster to reduce the number of records to be analyzed with a server query. The performance gain will depend on the relative complexity of the server query compared to the Manifold query. The more records that can be trimmed on the server side, the better the performance. Transforming data on the server (as opposed to trimming the number of records) will not gain anything.

The User

- Read the documentation and learn as much as you can so you are always using optimal methods.
- Take a few minutes to think about a task before launching into it. There are often many different paths to the same end within Manifold. A good plan will help you choose the best path and avoid unnecessary work.
- Don't fall into the trap of making projects more complicated than they need to be. Most GIS projects follow the 80 / 20 rule, where 80% of the desired result comes from 20% of the implementation effort. Do that 20% first and then try it out. You might find that you are happy with the result.
- Try to find a pre-existing data set that provides what you need before you invest time into building your own. Some people waste weeks of time digitizing raster scans to get a local map without realizing they could download 1:24,000-scale SDTS maps from USGS that provide the same thing for free.
- Try to find the data you need in modern formats that automatically convey all necessary information in a user-friendly way. Dealing with improperly used antique formats like shapefiles or non-GIS formats like DXF can be a huge waste of time if you can find the same data in a modern format allows instant, automatic usage. See Projections and Legacy Formats for a discussion of hassles that can be avoided.
- Become expert at using image servers to fetch satellite imagery on demand.
- Save your work regularly in separate projects so you never have to waste time redoing an entire project after a system crash or undoable user error at some stage.
- Memorize keyboard shortcuts and use them in combination with mouse moves. For example, experts will keep the left hand on the keyboard for CTRL-C and CTRL-V to Copy and Paste while the right hand moves the mouse to select items and manipulate windows and other mouse-based controls. The ALT-SHIFT keyboard shortcuts used to open and close Manifold panes are especially important if you have a small screen. See the Windows topic for tips on keyboard shortcuts.
- Learn to write scripts. Automating a task so that it takes care of itself while you are at lunch or at home is a wonderful time saver. At times a very simple script can replace a long sequence of commands using pre-built tools. When scripting, write the simplest code that works and use it. Write scripts so their internal functioning is obvious and include plenty of comments to boot. The objective is simplicity and maintainability.
- Sometimes you may be called upon to do a job under time pressure. The more time pressure you feel to complete a project, the more important it is to work systematically, steadily and carefully. Don't panic. Take it step by step in a steady pace. If you are short on time you don't have time for errors. Measure twice, cut once.
- Get plenty of sleep and exercise regularly. Fatigue causes errors and panic. Good health will help you think clearly and execute with authority.

People are sometimes amused that we include the user as part of our performance tips. However, the greatest gains in performance are usually achieved by using a better method or algorithm. More often than not the sole factor in whether a better method is used is the expertise and clarity of mind that can be mustered by the user. A healthy, well-rested, expert user is the best performance accelerator around.

General Education

Manifold works with many standard Microsoft facilities and products as well as with other industry standard products. Here are some examples drawn from many such features:

- Built into Manifold are the standard dialogs we would use to connect to data sources using ADO .NET, OLE DB and other such connection technologies.
- The Manifold Internet Map Server works with standard Microsoft Internet web serving technologies such as Internet Information Server (IIS), ASP and ASP .NET.
- We can write scripts in Manifold using standard Microsoft .NET languages or ActiveX scripting languages.
We can use standard Structured Query Language (SQL) to write queries.

Manifold can work with database products like SQL Server or Oracle.

One of Manifold's great strengths is exactly that Manifold uses a wide variety of industry standard products and technologies. Doing so makes it possible for people to apply in a familiar way within Manifold the expertise they have already acquired through the use of standard Microsoft products, and it also makes it possible for Manifold applications to leverage the power and flexibility of many other products and technologies in an industry-standard way.

However, this documentation does not teach such industry-standard products and technologies: it is assumed that if users want to employ such things they either already know how to use them or they have the ability to take advantage of the vast number of publications, web pages and training products that already exist for such industry standard products and technologies. There is no point in duplicating within this documentation those educational and support resources that already exist to a much greater degree.

Therefore, if we want to use a technology such as ADO .NET or ASP .NET or SQL Server we can certainly do so with Manifold. But if we don't know enough about such technologies to use them we should browse the web or go to our local high tech book store and buy one of the hundreds of books that teach the use of such technologies and then dig in and learn how to do so. Part of the point of using such technologies is exactly that they are already very well documented in a huge selection of publications.

Another benefit to using such standard technologies is that if we are unfamiliar with them and then have to learn about them to use them with Microsoft the education we invest into ourselves is not restricted to just Manifold. Learning about something like SQL Server or Oracle or ADO .NET will be useful knowledge in the many thousands of other applications that use such standard things.

**Systems**

Manifold is developed on Intel and AMD processors with average performance and not on unusually powerful machines. Most development is done using Windows Vista, Windows XP and Windows Server 2003. Faster machines are deliberately avoided to force a conservative perspective into development for core system functions. Given enough RAM for the task, required performance with almost any reasonably contemporary systems will be acceptable and performance on better than average systems will be stellar.

As processor speeds continue to improve performance on state-of-the-art machines will become dazzlingly fast. Machines with 4 to 8 gigabytes of RAM, many hundreds of gigabytes of disk drives, 64-bit quad core processors and CUDA-enabled nVidia graphics engines with over 100 stream processors have become very affordable. RAM, in particular, has become almost dirt cheap so there is no excuse for not equipping one's machine with gigabytes of RAM.

See the essay on Using RAM and other Machine Resources for the Manifold "spin" on this topic.

**Very Large Jobs**

No matter how fast Manifold can operate it will always be possible to ask Manifold to perform a task that will take a very long time to accomplish. Some jobs can seem to take an unreasonably long time if, without realizing it, we have suddenly increased the amount of data involved as compared to previous tasks.

Raster data and images can involve dramatically greater work for what appears to be a small increase in image size because the data in images increases as the square of any increase in height and width of the image. An image that's 40% larger takes 100% more work to manipulate. An image that is twice as big will take four times the work. For raster data such as DEMs it is easy to forget that scaling up from a 100 x 100 image to a 1000 x 1000 image will end up requiring 100 times as much work. What used to take one second can take 100 seconds with the larger image.

Many networking and geometric problems involve geometric growth in computation requirements. What appears to be a small increase in the problem can increase computation time from seconds to days. For example, growing the size of a road network so that it covers an entire state instead of a few counties could lead to a tenfold increase in the height or width of the map. The additional area of the map can easily add 100 times as many road nodes and links and thus lead to a million-fold increase in computation time for certain complex network tasks.

When performing computations that grow geometrically it is important to increase the size of the task in small steps. Begin by verifying your procedure with a very small subset of the data and then increase the size of the
problem in small steps so you can see where asymptotic growth in computation requirements begins. Note also that increasing the size of jobs will likely also place greater demands on system RAM.

As memory requirements increase, at some point a machine will run out of available RAM and begin paging to disk. At that point processing will become profoundly slower. See the Using RAM and other Machine Resources essay for why. To avoid this effect, use a machine that is better scaled to the task at hand or reduce the size of the job to what will fit into the machine being used. It is unrealistic to expect a machine with 128MB of RAM to be able to process large tasks as efficiently as one equipped with 8GB of RAM.

Some specialized tasks with large maps will take days to accomplish. People worldwide launch such jobs every day with Manifold using a spare machine that is left "cooking" for a few days to accomplish a desired task. They're happy because such things used to require weeks with older software or hardware. Run some experiments before launching such large jobs so you know what to expect.

Memory and Large Files

When Manifold opens a very large .map file not all of the contents of the .map are brought into memory at once. Components will be brought into memory as needed from the .map file stored on disk. Once they are in memory accesses will occur faster than in the initial usage since, of course, RAM memory is thousands of times faster than hard disk.

The advantage of having lots of RAM in a computer is that Windows editions such as Windows XP or Windows 2000 will leave items in memory until the memory is needed for something else by Windows. If we have ample RAM, as we work with a project the various components will end up in RAM and will stay there. This effect is especially pronounced with x64 Windows editions, which can actually work with large amounts of RAM effectively.

If you do not have enough RAM to run projects in memory, no matter how fast a processor you have you will have much less performance if the system begins paging to disk. That is why RAM is more important than processor speed until you have enough RAM so you never have to page to disk.

Notes

By mentioning certain brands we don't mean to imply anything negative about other brands. At this writing, the current generation of machines being installed at manifold.net use nVIDIA graphics engines and mostly Intel processors. nVIDIA has done a super job of writing good code for Windows and creating innovations like CUDA, which currently provides the fastest general-purpose processing around.

However, Intel and AMD's ATI divisions continue to be a major power in processors and no doubt will respond to nVIDIA's competitive stimulus with advances of their own. Likewise, we expect that NVIDIA's masterful driver work will inspire other processor vendors to pay better attention to writing effective drivers. Leadership positions in CPUs and graphics engines can change overnight, and often have.

From a software perspective, we love seeing the processor and graphics vendors work harder to offer more performance at lower price!

See Also

See the Memory Requirements topic for RAM and hard disk memory requirements.

See the Limitations topic for general notes on Manifold limitations when operated in various Windows systems.
Memory Requirements

Your computer's memory includes Random Access Memory (RAM) and hard disk memory. Manifold System uses both types of memory together with the Windows operating system and any other applications that may be running at the same time. Regardless of the amount of RAM or free hard disk space available, there are certain fundamental limits that cannot be exceeded:

- No single image or surface can exceed 16 EB (exabytes) in size. Since one exabyte is $2^{64}$ bytes, about a trillion gigabytes, as a practical matter there is no size limit for images and surfaces. One will run out of disk space billions of times sooner than the size limit is approached.
- In 32-bit Manifold editions, for components other than images or surfaces (drawings, scripts, tables, labels, comments), no single component can exceed 4 GB (gigabytes) in size. Projects are not limited in size, but no single component of these types in the project can exceed 4 GB. In 64-bit Manifold editions, no single component can exceed 16 terabytes in size.

The intrinsic limits on images, surfaces and other components are so large that they will likely exceed the processing capacity and disk storage capacity of most machines in common use.

Use x64 Windows and x64 Manifold Editions

This is so important it is worth saying right up front: 32-bit Windows editions are crippled by an inability to use more than a relatively small amount of RAM. RAM has become so inexpensive that there is virtually no excuse not to have several gigabytes, at least 4 GB, of RAM in your computer. However, to use that RAM effectively you must be running x64 Windows and x64 Manifold. x64 Windows is more reliable and runs much faster and enables use of x64 Manifold, which runs much faster than 32-bit editions of Manifold.

Hard Disk Free Space Requirements

There are four demands for hard disk free space made by Manifold:

- **Project files** - There must be enough free space on hard disk for the size of project (either compressed or uncompressed) when it is saved as a `.map` file. Obviously, if we have only 500MB of free space on disk we will not be able to save a `.map` file that is 1 gigabyte in size.

- **TEMP files** - When Manifold opens a `.map` file it creates one or more temporary copies in the Windows TEMP folder to allow abandoning edits (that is, to close a project without saving any changes), to enable operation of Undo and for other purposes. Such temporary files can temporarily require four times the size of the project in uncompressed form. Thus, to work with a 1 gigabyte project in all circumstances we should allow at least four gigabytes of free space on hard disk.

- **System paging file** - Microsoft Windows automatically uses virtual memory by swapping processes out to a hard disk file called the system paging file or pagefile. If we have only 512MB of RAM but want to run applications that require 1.5 gigabytes or RAM, Windows will automatically swap parts of the executing processes back and forth between RAM and the paging file so that at any one instant only 512 MB of RAM is used. There must be enough free space on hard disk for as large a pagefile as is necessary for Manifold and any other applications that are running.

Some users have multiple disk drives or have organized a single disk drive as multiple volumes. Make sure you have enough free space on all disk drive volumes involved if the project files, TEMP files and paging file are on different disk volumes. For example, a user might think there are tens of gigabytes of free space available on an E: disk volume where project files are kept without realizing that there is no free space on the C: disk volume where the Windows TEMP files and paging file might be kept.

- **Cache files** - If you work with linked images from OGC WMS servers or TerraServer, Manifold by default will cache images and parts of images. **Caution:** Cache files persist forever and, in extreme cases, use up your free disk space unless they are manually deleted. See the recommended actions in the Managing Cache Files topic.

RAM and Virtual Memory Requirements

For performance reasons it makes sense to have as much RAM as possible and to use x64 Windows and x64 Manifold so you can take advantage of that RAM. It is a good idea to have somewhat more RAM than the maximum project size to allow RAM for Windows and for other applications that may be running in background.
To work with large projects, one must have a system pagefile that can expand to as large a size as is necessary and the free disk space available to host large pagefiles if need be. In addition, it is critically important to have enough free disk space for all TEMP files that must be created by Manifold.

**Very Important:** When working with large projects make sure to set your Windows use of virtual memory appropriately. Some editions of Windows, such as Windows XP x64, will set a maximum pagefile size that is too small by default. Drill down through the Control Panel - System applet's Advanced settings to change the maximum pagefile size to four or five times the largest project you will need to use. For example, if you will be working with 10 GB projects set the pagefile size to 50 GB.

Geometric computations and many other functions use virtual memory for temporary computational requirements. Such virtual memory will be automatically switched by Windows between RAM and disk but can never exceed 2 GB in 32-bit Windows editions. This requirement for internal computational storage space means that while components and projects can be very large, the temporary storage space limitation of 2 GB may prevent some computational operations from proceeding to completion with very large data sets. For example, when creating buffer zones each buffer zone must be able to fit into Windows virtual memory. Running the Buffers transform operator on a drawing therefore will not work unless the buffers to be created will require substantially less than 2 GB for the buffer zone objects.

Although 2 gigabytes, of course, is enough for an immensely large number of buffer zone objects it may not be enough for, say, creating a buffer zone around every object in a 3 or 4 gigabyte drawing. In such cases, divide up the job into smaller parts and perform each in turn.

Users working with 32-bit Windows should keep in mind that the 2 GB process space will not all be available for a particular task. If you are using a 32-bit edition of Manifold, the Manifold process can only use 2 GB of memory. Some of that memory is spent on system services as well as Manifold services such as the window system. This can leave only about 1 GB of memory for tasks such as Normalize Topology or running the Buffers transform operator.

Memory usage patterns vary greatly between various algorithms used within Manifold, but a lack of memory can be encountered if an algorithm has used up some memory (for example, 500 MB) and wants more (for example, 1000 MB), but the system can not provide it. Thus, a times a lack of memory error can occur even though the amount of memory in actual use is well below the limit.

The way to avoid such difficulties is to utilize 64-bit Windows with x64 Manifold editions and to install plenty of RAM. x64 Manifold editions can use 16 Terabytes of memory, the same as can be addressed in 64-bit Windows.

**Saving and Loading Projects**

Manifold will often encounter peak demands for memory when opening or saving a project because it is at that time when reserve copies of projects are being made or are being utilized to allow abandonment of imports or exports or edits, or to Undo operations. When operating with limited amounts of free disk space it is possible for disk space requirements to exceed the amount of free space that is available on disk.

Free hard disk space available might change between the time a project is saved and when later it is opened. For example, we may have created a very large project and saved it and then later used up free space on our hard disk so that little or no free space is available. In such cases we might not be able to open large projects that were previously saved because we no longer have sufficient free space on disk for the TEMP files that must be created when the project is opened.

**Reducing Memory Usage**

When Manifold performs various operations it may need to use memory (either RAM memory or temporary files on disk) to allow Undo or abandonment of the project when exiting without saving changes. Saving the project will free memory that has been occupied by changes made since the last File - Open or File - New operation. Such savings can be very important when working with large amounts of data.

In addition to the suggestions in the Performance Tips topic, reducing the amount of data in tables can reduce memory requirements. At the very least, make sure to convert fixed-length text columns to variable-length text columns. Another space-saving option would be to store data in a linked table (say, within SQL Server) and to map the columns from the linked table to a drawing's table via relations.

**A Note on .map File Compression**
Saving .map project files in compressed form will save a lot of space on disk for routine storage of .map files. However, it does not save any space in TEMP file usage or other memory requirements because all operational work is done on the uncompressed data. The relatively small size of compressed .map files may catch users unaware as their files expand, possibly by a factor of ten, into operational use.

For example, a compressed .map file may be 200 MB, but it could easily expand by a factor of seven when uncompressed into a 1.4 gigabyte project. A 1.4 gigabyte project could require almost six gigabytes of free space on disk for temp files. If we planned on needing only 800MB of free disk space based on the compressed size of the project, we might think we had plenty of free disk space if we had four gigabytes free space available. However, that would be too little to deal with a project of this size once it is uncompressed and in active use.

**Tech Tip**

If for any reason (such as an electrical power failure) you crash Windows or crash Manifold by attempting to work with very large projects without enough free disk space you could end up with some large files in your Windows TEMP folder. Delete these before continuing work so that you do not waste disk space on unnecessary TEMP files.

**See Also**

See the Performance Tips topic for general advice to squeeze the most performance and capacity from your computer system.

Manifold System is designed to take advantage of modern computer systems, which often feature lots of inexpensive memory. See the Using RAM and other Machine Resources essay for information on the technical tradeoffs between using RAM and other resources.

See the Limitations topic for general notes on Manifold limitations when operated in various Windows systems.
Managing Cache Files

When downloading image information for linked images created from OGC WMS image servers, Manifold Image Servers or from TerraServer servers, Manifold provides a cache data option (on by default) to cache image files that are fetched on local hard disk. To prevent cache files from growing without limit, users must clean out old cache files from time to time.

The Cache data option is important because it keeps a copy of the linked image in local disk cache. If this option is on, the next time we open a project containing a linked image, Manifold will be able to use the copy of the image saved in the local disk cache and will not have to go through the process of downloading the image again. The data cache will persist across sessions and projects so that once a tile is downloaded it will not have to be downloaded again. If we have a large hard disk with many gigabytes of storage available we can cache large numbers of large images.

The location of the cache folder is at whatever location is specified in the Data Cache item in the Tools - Options - File Locations dialog, with WMS cache files being stored in a series of subfolders and TerraServer cache files being stored in a TerraServer subfolder. The default value for the data cache location is %MyDocuments%, so by default each user's files will be stored in the user's own cache folder.

When multiple users are working it might make sense to modify the default Data Cache setting for each user so the cache folder location is the same for all users and is on a shared resource available from all machines on the local network. For example, if there is a machine called Storage on our local network with a Cache folder on its C: drive, we might use a Data Cache setting of \Storage\Cache for all users. In that case, any tile files that are retrieved by one user will be available to all users so that tile files for a given view will never have to be fetched twice over the network. This assumes, of course, that the network is fast enough so that use of cache over the network will not be too slow.

It is also important when working with large linked images that the location specified for the cache is on a disk drive with plenty of free space. Users sometimes will ask for download of very large linked images without thinking about how large the resultant image may be and thus end up inadvertently requesting immensely large data sets.

The data cache folder may be moved to a different location so long as the path specified in the Data Cache option is adjusted to point to the new location.

Cleaning Out Old Cache Files

Cache files persist forever until they are manually deleted. Manifold will not clean out cache files or automatically delete older cache files because cache files may be in use by multiple users if they are in shared storage. To avoid running out of free disk space, users should manually delete unnecessary cache files from time to time.

To delete unnecessary cache files, using Windows Explorer browse over to the location specified in Tools - Options - File Locations for data cache files. Set Windows Explorer so it shows file details, including the date each file was last modified and then right click on the Modified column in Windows Explorer to sort the files by the date of their last modification. Delete those files that haven’t been used for a long time to free up disk space.

See Also

Linked Images from OGC WMS Servers
Linked Images from TerraServer
Tools - Options - File Locations
Using the Manifold ODBC Driver

Installing Manifold automatically installs the Manifold ODBC driver that allows other applications to access tables and queries in Manifold .map project files. Manifold's ODBC driver includes support for OGC clients. Linked and shared components (components used in Enterprise Edition) are handled transparently. The driver uses the internal Manifold SQL engine and supports intrinsic columns as well as spatial, geocoding and other extensions available within the main system.

The Manifold ODBC driver operates in read-only mode and has been tested to work with various versions of Microsoft Office applications such as Access and Excel, the Microsoft OLE DB Provider for ODBC, a wealth of various development tools and utilities and, of course, Manifold System itself. The Manifold ODBC driver supports optional ODBC functionality required by MFC clients as well as optional ODBC functionality required by the latest editions of Visual Studio.

The Manifold ODBC driver may be used in Microsoft Office data sources. This makes it possible to have a .map file that includes a table linked from another .map file.

Note that because the Manifold ODBC driver is read-only, the integrity checking features of Access aimed at doing consistent updates are not supported when Access connects to a .map file through the Manifold ODBC driver.

Configuring a Data Source

The first time Manifold is launched it will automatically install the Manifold ODBC driver. This will add the driver to the Drivers tab of the Data Sources (ODBC) Control Panel applet (located under Administrative Tools in Windows 2000 and Windows XP).

To add a data source, go to the User DSN page within the Data Sources (ODBC) applet and click Add. Select the Manifold Project Driver and click Finish.

In the Manifold ODBC Driver dialog type the name to use for the data source, for example, “My Manifold Connection” and select the .map file to be used for the database. Choose the options desired:

- **Name**: The name by which we want this connection to be known.
- **Database**: The name of the .map file to be used.
- **ANSI-compatible queries**: Use ANSI syntax for queries executed against this data source.
- **Translate Unicode data to ANSI**: Convert Unicode text to ANSI text on the fly. Checked by default. Required for use with applications that do not support Unicode columns.
- **Expose data for OGC SQL clients**: Expose WFS for SQL used by OGC clients.

Click OK. We can now use the created data source from within other applications. We can also use it within Manifold’s own Data Source dialog. We can also add data sources using the ODBC applet’s System DSN and File DSN pages if desired.

To configure an existing data source, locate it within the Data Sources (ODBC) applet and click Configure. We can then make changes, such as changing the path to the .map file or changing the status of the Translate Unicode data to ANSI option and click OK to accept changes.

To remove an existing data source, locate it within the Data Sources (ODBC) applet and click Remove.
Components linked into an Enterprise Edition project are said to be **shared components**. Shared components may be **cached** or **uncached**, as described in the Cached and Uncached Components topic. When working with the Manifold ODBC driver from another application, keep in mind the following two rules:

- The Manifold ODBC driver works only on machines on which Manifold is installed. For example, if machine A has Manifold installed but machine B does not have Manifold currently installed, then an application running on machine A will be able to work with Manifold .map files on either machine A or machine B using the ODBC driver. However, it will not be possible for an application on machine B to work with .map project files stored on either machine A or machine B.
- The license status of the local Manifold installation specifies the degree to which the ODBC driver can access uncached Enterprise server components. For example, if machine A has Manifold Enterprise Edition installed and machine B has Professional Edition installed then machine B will not be able to access uncached shared components stored in .map files on machine A.

Accessing a .map file containing shared components from the Manifold ODBC driver is analogous to accessing the .map file from a local Manifold System application on this machine. The Manifold ODBC driver works by utilizing the facilities of whatever Manifold System program has been installed on the machine so the ODBC driver has the same capabilities as running Manifold System interactively on that machine.

Shared components that are cached may be accessed like all other components. Shared components that are uncached will be transparently fetched from the Enterprise server or will appear empty depending on whether the machine has an Enterprise Edition license. If the machine has an Enterprise Edition license and the Get latest versions of shared components after opening file option is on for the current user, loading the .map file through the Manifold ODBC driver will also fetch the latest version of shared components that are cached.

**OpenGIS Interaction**

The Manifold ODBC driver includes an **Expose data for OpenGIS clients** option that makes the ODBC driver mimic an OpenGIS data source. When the option is turned on, the ODBC driver creates several virtual tables for each drawing in the MAP file on which it is launched so that OpenGIS-aware clients can link to them. By default, the option is turned off.

This option is especially useful for OpenGIS clients that can consume OpenGIS data via SQL, such as TatukGIS. OpenGIS users may also be interested in reading the **WKB** information in the Geometry in Tables topic.
Command Line Options

Manifold provides several command line options when launching Manifold from a command line:

/activate:serial,key  Activate Manifold System.

/activate:<invalid serial>  Pop open the Activation dialog when an invalid serial number is supplied with the /activate option.
Example:

manifold.exe /activate:dialog

will launch the Activation dialog since "dialog" is not a valid serial number.

/activateExt:serial,key  Activate an extension.

/autoexec:<scriptname> <mapfile>  Run the specified script after opening the MAP file specified.

/cfilter:<path>  Launch Manifold using a command filter.

/clist:<textfile>  Dumps the names of all commands available for user interface scripting using the InvokeCommand method of the UserInterface object into a text file. Example:

manifold.exe /clist:file.txt

/logquerytime  Logs the running time of queries into the History pane.

/logrenderingtime  Logs rendering time for component windows into the History pane.

/logscripttime  Logs execution time of scripts into the History pane.

/logtransformtime  Logs time for drawing, image, surface and table Transform Toolbar operations into the History pane. Also logs execution time for the Surface - Transform dialog, spatial overlays and topology overlays in addition to Transform Toolbar operations.

/slist:<textfile>  Dumps the names of all formatting styles available for areas, labels, lines and points into a text file. Example:

manifold.exe /slist:file.txt

The examples shown above for /clist and /slist options will create the specified file.txt file in the same folder as the manifold.exe file, the installation folder used for Manifold.

Those examples will not work in Windows Vista if Manifold has been installed in the default installation folder within the C:\Program Files hierarchy, because Vista in general does not allow writing to such folders by default by ordinary user logins.

In such cases specify a file location that is known to be writeable by the login in use. For example, if we have a folder called C:\tmp that we know is writeable we could use

manifold.exe /slist:C:\tmp\file.txt

Command Customization

Manifold may be customized to control what commands and menu items are available using command filters.
Launching Manifold with the /cfilter:<path> option starts Manifold using a command filter. For example, entering the following at the Windows command prompt (within the installation folder for Manifold) or within a batch file will launch Manifold using a command filter:

```
manifold.exe /cfilter:\myfiles\customizations\mymanifold.xml
```

The above will use the mymanifold.xml command filter file located in the \myfiles\customizations\ folder. See the Command Filters topic for details.

### Scripting

Manifold provides sophisticated scripting to extend the capabilities of the program.

The /autoexec:<scriptname> <mapfile> command line switch allows running a script after opening the MAP file specified in the same command line. For example, the following command line will open the mymap.map project file and then run the myscript script found in that project file.

```
manifold.exe /autoexec:myscript c:\MyProjects\mymap.map
```

See the Scripts topic for details on scripting. See the User Interface Scripting topic for a quick and easy interactive method of discovering command names usable when scripting the user interface.

### Activation

Manifold includes a security system for license authentication that is described in the Activation Keys and Serial Numbers topic. See that topic for a discussion of command line options for activation.

### Formatting

New releases of Manifold may introduce new formatting styles. The /slist command line option provides a handy way for programmers to get a comprehensive list of all area, label, line and point styles available and their names.

### Technical Note

Strictly speaking, it is possible even in Vista to run commands of the form:

```
manifold.exe /slist:file.txt
```

Launching the command prompt window using the Run As Administrator command will allow writing to C:\Program Files even in Windows Vista. If the path to the Manifold installation folder is cited in the PATH variable, the above command can be executed in the context of any folder. However, this is a Vista nuance not related to Manifold so to avoid problems for users who do not have access to Administrator passwords this topic offers the advice to use folders that are known to be explicitly writeable by using commands of the form:

```
manifold.exe /slist:C:\tmp\file.txt
```

### See Also

Scripts
Customization
Command Filters
Activation Keys and Serial Numbers
View - Panes - History
For Experienced GIS Users

This topic introduces Manifold to experienced GIS users. It assumes you have a lot of experience working with previous Manifold, ESRI or MapInfo GIS products.

This topic provides a brief survey of Manifold that may allow GIS experts to more efficiently digest the rest of the documentation. After reading this topic, experts should continue to read other topics, even introductory ones, in the suggested order, beginning with the Introduction.

The minimum amount of reading, even for experts, is to read all of the topics in the Introduction chapter as well as all of the topics in the Examples chapter at the end of the Help file. Manifold does many things different than older GIS packages, so even if you know GIS very well do not skip the introductory reading.

Manifold Editions

There are several versions of Manifold System available for licensing as well as optional extensions. Manifold System versions include:

- **Personal Edition** - Includes almost Manifold System features in this documentation, except the Internet Map Server, Database Administrator Edition features and Enterprise Edition features.
- **Professional Edition** - Adds the Manifold Internet Map Server (IMS).
- **Enterprise Edition** - Adds centralized document-oriented storage of components within Enterprise database servers. Adds direct support for object-level storage in Oracle Spatial databases as well as Oracle GeoRaster support. Enterprise Edition also eliminates the requirement for a Manifold logoype in IMS pages and provides the ability to save to .e00 format. See the discussion in the Data Storage Strategies topic for various server storage options available to Enterprise Edition users.
- **Universal Edition** is a product configuration that contains an Enterprise Edition license plus Business Tools, Geocoding Tools and Surface Tools licenses.
- **Database Administrator Edition** is a version of Enterprise Edition that includes the Administrator Console and the Batch Export tool and is intended for use by DBMS administrators. It is used to configure database systems for use with Manifold System when saving drawings or other data in the DBMS to enable concurrent multi-user editing. Administrator Console enables use of friendly names, formatting and other features that once set up via Administrator Console can be used in ordinary Enterprise Edition licenses. Larger Manifold installations will employ mostly Enterprise Edition or Universal Edition licenses for users and will have a few Database Administrator Edition licenses for use by IT managers and DBMS administrators.
- **Manifold System License Server** gives organizations the ability to maintain an inventory of floating Manifold licenses that may be used by any client computer that can connect to the License Server, all without needing to use serial numbers and Activation keys for individual client licenses. License Server provides the economy of floating licenses as well as greatly reduced administrative overhead for organizations that deploy many hundreds or thousands of Manifold licenses.

The Business Tools package is an optional extension to Manifold System that provides additional commands for working with drawings. The extension adds functionality in several areas:

- **Districts (Advanced)** - Redistricting using a condensed, expert level dialog.
- **Districts (Visual)** - Redistricting via an easy-to-use visual, interactive dialog.
- **Drive-Time Zones** - Computation of drive-time zones via an easy-to-use, interactive dialog.
- **Optimal Route** - Finding an optimal route to designated locations using a condensed, expert level dialog.
- **Optimal Route (Visual)** - Finding an optimal route to designated locations via an easy-to-use, interactive dialog.
- **Send Email** - Automatically send email to map objects. Used to send spatially-targeted email.
- **Topology Factory** - An interactive dialog to view and repair common topological errors in drawings.
- **Expanded programmatic access to routing and other functions.**

If you do not have the optional Business Tools package enabled you will not have the above capabilities enabled within Manifold System.
The Geocoding Tools package is an optional extension to Manifold System that provides street address geocoding capability as well as access to the Manifold Geocoding Database data provided on the Manifold downloads site for both US street address geocoding as well as the easy creation of drawings showing US streets. The extension adds functionality in several areas:

- Street address geocoding within the United States using the Manifold Geocoding Database provided on the Manifold downloads site.
- Street address geocoding within North America or Europe using the appropriate edition of Microsoft's MapPoint product as a data source.
- Street address geocoding anywhere in the world using user-provided geocoding data extensions in either range or points of interest formats.
- Street address geocoding using web-based Manifold geocoding servers to provide geocoding services.
- Easy import of drawings showing US streets from the Manifold Geocoding Database provided on the Manifold downloads site as set forth in the Import Drawing - Geocoding Database topic.
- Geocoding Extensions may be used within SQL to allow use of street address geocoding functions within SQL queries.
- Programming objects such as the Geocoder object may be used from scripts or other programs or from within IMS applications.
- Street addresses and ZIP codes may be used in the Edit - Go To dialog.

If you do not have the optional Geocoding Tools package enabled you will not have the above capabilities enabled within Manifold System.

The Surface Tools package is an optional extension to Manifold System that provides additional commands for working with surfaces. The extension adds functionality in several areas:

- The Surface - Transform command dialog allows arbitrary transformation of surfaces, including computations that involve multiple surfaces such as subtracting one surface from another. A very rich collection of operators allows many different types of tasks to be accomplished.
- The Surface - Watersheds command works with a surface and optionally a drawing to find watersheds in the surface (regions sharing a common drainage) as well as streams in the surface or upstream areas in the surface from points in a specified drawing.
- The Transfer Heights command enables rapid transfer of surface values from surfaces to points.
- The Visible Area command shows areas that are visible from given points. The height of viewing points may be automatically adjusted from a height field.
- New options in surface generation allow creating surfaces from drawings and tables using Gravity interpolation, Median-Polish Kriging and triangulation. See the Project Pane - Paste as Surface topic.
- Expanded model choices for Kriging interpolation include Linear, Power, Rational and automatic model choices.
- New transform toolbar operators for interpolation of missing pixels, Interpolate, Interpolate (Parameter) and Interpolate Row.
- Surface Tools adds the ability to work with new Profiles and Elevations components that show cross-sectional cuts through a given path over a surface.
- Expanded programmatic access to surface operations.

If you do not have the optional Surface Tools package enabled you will not have the above capabilities enabled within Manifold System.

This documentation is written so that it may be used with all editions of Manifold System and with all options. Topics that are unique to a particular edition or option are so marked.

Manifold is Different

Manifold's user interface is significantly different from ESRI or other traditional GIS products. Manifold's user interface is based primarily on Microsoft style as well as user interface standards that are prevalent in professional quality, mass-market applications. Using a more widespread interface style opens GIS to many more people who are already familiar with mainstream applications like Microsoft Office.
Although the user interface is different, experienced GIS people will be able to re-cycle a lot of conceptual ideas. For example, the experienced reader knows what a projection is and that projected coordinates are linear measures, not degrees. Please, however, do not let this general familiarity with the subject matter mislead you into attempting to use the product without reading the documentation.

Even if you consider yourself to be a Windows power user, please read the Windows topic. This describes key Windows and Manifold power moves. It’s wise to learn keyboard shortcuts for opening and closing panes, and to learn how to hot scroll.

Manifold makes heavy use of Copy and Paste As to convert information into different forms. For example, we Copy a geocoded table and then Paste As a drawing to create a drawing from the geocoded table. Labels components are created that show fields by copying a drawing and pasting it as a labels component. Images may be pasted as tables. Please take time to review the Copy and Paste As topic for more on this essential method.

Note: If you’ve Pasted something into a drawing and don’t see it, try Zoom to Fit to make sure it is not outside the window at the current zoom and pan setting.

Terminology

Manifold uses mass-market terminology to refer to points, lines and areas. This is what most of the world’s mathematics and computer community uses. Some traditional GIS packages, such as ESRI’s, use different terms:

<table>
<thead>
<tr>
<th>Manifold</th>
<th>ESRI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Areas</td>
<td>Polygons</td>
</tr>
<tr>
<td>Lines</td>
<td>Arcs</td>
</tr>
<tr>
<td>Points</td>
<td>Points</td>
</tr>
<tr>
<td>Surface</td>
<td>Grid</td>
</tr>
</tbody>
</table>

The above are non-trivial differences in terminology that affect GIS users everywhere due to the prevalence of ESRI terminology and its effect on traditional GIS nomenclature. See the essay titled Terminology in GIS for why Manifold uses the more popular terms. Manifold also replaces some traditional GIS words with words that are more prevalent in computing:

<table>
<thead>
<tr>
<th>Manifold</th>
<th>Traditional</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image</td>
<td>Raster data</td>
</tr>
<tr>
<td>Drawing</td>
<td>Vector data</td>
</tr>
<tr>
<td>Tracing</td>
<td>Vectorization</td>
</tr>
<tr>
<td>Coordinate</td>
<td>Inflection point, or Node</td>
</tr>
<tr>
<td>Network</td>
<td>Graph</td>
</tr>
<tr>
<td>Node</td>
<td>Vertex</td>
</tr>
<tr>
<td>Link</td>
<td>Edge, or Arc</td>
</tr>
</tbody>
</table>

Manifold follows accepted modern network terminology so that the English language versions of Manifold use the word “network” instead of “graph,” “nodes” instead of “vertices” and “links” to mean the connections between nodes in a network.

Although in casual usage most GIS systems refer to the files they produce as “maps”, Manifold uses this word to refer to a specific type of component that contains other components in layers and which may be re-projected on the fly. Manifold uses the word “drawing” to refer to a vector item and “image” to refer to a raster item.
Manifold uses the word theme in a somewhat different way than ESRI. In Manifold, a **theme** is a project component that shows a drawing with the formatting specified by the theme. A drawing can have many different themes, each of which shows the same drawing using different formatting.

**Advice to ArcInfo Users**

Manifold takes a different structural approach than ArcInfo that requires some conceptual adjustment. Unlike older GIS packages, Manifold does not build topological relationships into the data set: rather, topological relationships are computed on the fly as needed. There is no need for users to worry about maintaining topology within drawings and maps in the low-level way that was done earlier. Manifold will do this for you automatically.

Years ago when computers were weak it made good sense to embed topological relationships into the data sets themselves. In modern times it is better to compute topology on the fly. Doing so uses the intelligence of the machine to free the user from having to worry about low-level structural details. It allows us to mix points, lines and areas in the same drawing and to move objects about interactively with great freedom.

We ask ArcInfo users to approach Manifold without worrying about the details of how something is done internally and to just take it on faith that Manifold will compute topological relationships correctly when needed. We suggest a focus on the higher levels of function while learning the system.

For ArcInfo users who would like to experiment with more familiar topology methods within Manifold, the Traditional Topology Tools optional add-in may be downloaded from the [manifold.net web site](http://www.manifold.net). This add-in provides operations using stored topology.

**Accuracy**

Manifold maintains all geographic geometric information using double-precision floating-point numbers. This provides accuracy better than 1/250,000,000 of a meter (literally, less than a two hundred fifty millionth of a meter) at the Equator.

If you do not need very high accuracy, you can increase the speed of many operations by reducing the location precision of a drawing using the View - Properties dialog.

**Files and Project Organization**

Manifold's documents are **projects**, which are saved in a .map file. Projects consist of **components** like drawings (vector data like ESRI shape files), images (raster data, like a GeoTIFF), surfaces (typically, terrain elevation data like a DEM), tables (data attributes for drawings or simply independent database tables) and other items. Create a new project with File - New.

The project is a single Manifold .map file. The project .map file by default contains in one file all of the vector, raster and other data imported into the project. Keeping everything in one file makes it easy to keep track of what's in a project and to share it with other users. Send them the .map file and they have everything.

There are, however, many circumstances when users would like to keep some data outside of a project. We can accomplish that in Manifold by using **linked** or **shared** components. **Linked components** include:

- Linked tables and queries - tables or queries can be either imported into the project or they can be left as external tables or queries that are dynamically linked into the project.
- Linked drawings - drawings can be linked into a project from external data sources, either from geometry columns in tables or queries or from geocoded tables.
- Linked images - images can be linked into a project from compressed image formats, from external databases or from a variety of image servers.

**Shared components** are components saved in a Manifold Enterprise Server and used within an Enterprise Edition project. Enterprise Edition allows saving components within Enterprise servers set up within a database server, which may then be linked into user projects on different machines. Enterprise Edition allows control over whether shared components are cached or not cached. If they are not cached, they will take up no space in the local project but rather will be stored only on the Enterprise server.
See the discussion in the Data Storage Strategies topic for an overview of local, shared or server data storage options available to Manifold users.

In the addition to the above, Manifold supports sophisticated storage of data in enterprise-class databases such as Oracle Spatial. Manifold can use Oracle Spatial, IBM DB2 with Spatial Extender, PostgreSQL / PostGIS or Microsoft's SQL Server 2008 (Katmai) with native spatial DBMS functionality.

In addition, Manifold can confer true spatial DBMS functionality unto almost any DBMS, such as MySQL or regular SQL Server Express. When using generic Manifold spatial DBMS capability, Manifold can store not only drawings but also images and surfaces into the spatial DBMS.

Manifold's extensive abilities to work with external DBMS packages sometimes fools people into thinking that Manifold requires some external DBMS to function. That's not true, as Manifolds' own .map project files provide efficient file-based storage. A Manifold .map project file can store all components of a Manifold project, including images, surfaces, drawings, labels, scripts, comments, queries and other components.

Some people think it is wasteful to import everything into a single file. In an era of nearly free hard disk storage it's no big deal and greatly simplifies exchanging maps: no more scrounging around to see if you remembered all the files you need to make your map. People who disagree are always free to use linked components or Enterprise Edition so that all components can be stored on a centralized Enterprise server.

Except for the case of linked drawings linked from external data sources like Oracle Spatial databases or other geometry types, Manifold does not edit external files "on the fly" in their native formats. For example, we don't open a shapefile in Manifold and then edit it. Instead, we import the shapefile into one or more Manifold drawings, do what we want, and then if we are cursed with having to use shapefiles we can always export the drawing out to shapefile format again.

The reason Manifold does not edit shapefiles or similar formats "in place" is that such formats are too primitive to support the full range of Manifold functions. Editing such files within the limits of what the format can support would require dumbing down Manifold functions or otherwise restricting the Manifold user interface to no more than can be accomplished with such primitive formats. That doesn't make sense: instead, it is better to use the full facilities of Manifold as desired and then, upon export, explicitly do whatever dumbing down is required to fit the target format.

**Note:** The three-letter .map file extension may be used by other applications on your computer. When you install Manifold it will automatically tell Windows to open .map files with Manifold. If you would prefer to retain the assignment of files ending in .map to some other application, in Windows Explorer right click on a .map file and choose Open with and then Choose Program. Choose the program you wish to use to open .map files and check the box Always use this program to open these files and press OK.

**Basic Steps to Running Manifold**

- Launch Manifold and note that the project pane is open (it is by default and can be toggled on/off with SHIFT-ALT-P).
- Import the drawings to be used. ESRI users will import .shp or .e00 files, for example. When importing from .shp you may need to set the datum manually, since shapefiles do not save projection info. See the Import a Shapefile example.
- To show drawings in layers, create a map that contains those drawings as layers.
- To change colors or otherwise format, click on the layer to be formatted and change settings in the format toolbar. If you would like to show a drawing using different formats, use a theme to do so.
- To print something, either use File - Print for casual printing or create a layout for more sophisticated printing and then print the layout.

**Importing Files**

Import vector data into drawings or raster data into images by clicking File - Import. There are importers for every GIS and raster image format in common use as well as lots of uncommon formats. If there is a format you need that is not currently supported send a note to sales@manifold.net so we can consider adding it to the release in a Service Pack.

Whenever a drawing is imported from a GIS format like MapInfo .mid/.mif or ESRI .shp that has both geometric data and data attributes in the format, both a drawing and an associated table will be automatically created.
Please read the specific importer topic for the file format you are importing within the Import and Export section to see how Manifold handles details for your format.

This topic mentions “drawings and images” as a shorthand way of referring to typical components. Other components are surfaces and labels components. However, since “drawings, surfaces, images or labels” is a bit much we shorten the phrase at times.

**Importing Projected Drawings, Surfaces or Images**

Manifold automatically will import projection information from those GIS formats that save it. When importing projected drawings or other components from formats that do not save projection information, we will have to provide the missing information manually.

As an experienced GIS user, you know it is a bad idea to save projected data in legacy formats like AutoCAD .dx or ESRI .shp because such formats as commonly used do not save the projection parameters that are necessary to make sense of the data. If you encounter projected data saved in such formats, keep an eye out for any accompanying documentation that describes the projection parameters used.

To import projected data from a legacy format:

1. Import the drawing, image or surface using File - Import. Use default settings.
2. Open the component thus created.
3. Use the Edit - Assign Projection dialog to specify the projection information that should be used.

The Edit - Assign Projection dialog allows us to manually provide the correct projection information into the coordinates properties. Manifold will then be able to make sense of the data just imported.

Do not confuse the use of a chosen projection view in a map window (via the Edit - Assign Projection dialog) with the use of Edit - Assign Projection in a drawing window to specify projection information missing from legacy formats. If a projected drawing is imported from, say, .shp format and you fail to tell Manifold the correct projection to use, that drawing has not yet been correctly imported. Subsequent use of the drawing in maps may cause bizarre effects or lengthy delays as the system attempts to compute a new projection for the map based on fundamentally inaccurate coordinates.

**Default Use of Orthographic**

Every drawing and image in Manifold exists in some geographic context. This is so even if we intend to use Manifold as a CAD or image editor to work with drawings and images in non-geographic contexts. By default, files such as .dx that are not in latitude / longitude degrees are imported as if each unit was equivalent to one meter. The scale factor can be changed during import or at any time afterwards. The file is imported into Orthographic projection and positioned with its lower left corner at the intersection of the Equator and Prime Meridian (zero longitude and zero latitude).

When importing images from geographically mute formats such as .jpeg or .bmp the image is automatically georeferenced to the (0,0) lat/lon origin using Orthographic projection as well. Manifold is so accurate that each pixel of a 4800 DPI image can have its own distinct lat/lon geographic location. Of course, if an image originates in a geographically aware format such as GeoTIFF it will automatically be georeferenced to the correct location. Images may be georegistered using drawings or other images that are already georegistered.

**Re-Projecting a Component**

We can permanently change the native projection of drawings, images, surfaces, or labels by opening the component and then using the Edit - Change Projection dialog to change the projection. See the Edit - Change Projection topic.

**Very Important:** Be careful not to use the Edit - Assign Projection dialog to attempt to change an existing projection. The Assign Projection dialog is used when importing projected data from legacy formats such as .dx or .shp, to specify the projection that is to be used. To change the projection of a component use the Edit - Change Projection dialog.

**Maps**
A special type of component is a map. A map shows drawings, images, surfaces and labels components in layers. After importing the various drawings and images we need into the project we create a new map by using **File - Create - Map** to add it to the project. The **File - Create - Map** dialog will offer up a list of components we can include in the map.

Double click on a component in the project pane to open it in its own window. When we first open a map we'll see it uses the projection of one of the components used to create it. When a map is first created, it will use the projection of the largest image or surface used to create it. If there are no surfaces or images in the map, the map will be created using one of the projections of the drawings it contains. At any time we can change the projection used by the map, although it makes sense to use the projection of one of the large components in the map to avoid any performance loss resulting from the need to re-project components on the fly.

Use **Edit - Assign Projection** to specify what projection the map should use. Maps can use whatever projection we like even if the drawings and images they contain use different native projections. Changing the projection used by a map does not change any data or the native projection used by any drawings or images the map contains. Projections shown by a map are computed "on the fly" without any change in the original data. We can change them however often we like without any decrease in accuracy.

There are two ways to change the projection used by a map:

- Use the **Edit - Assign Projection** dialog to set any projection
- If we want the map to use the native projection used by one of the components it contains, right click on that component's layer tab and choose **Use Projection** from the context menu that pops up.

Add drawings, images or labels components to a map by dragging and dropping them from the project pane into an open map window.

Control layers in a map by either dragging layer tabs in the map window or by using the **View - Panes - Layers** pane. Right click onto layer tabs for useful context menus. The Layers pane is much more useful than the layer tabs when there are many layers in a map. Each layer is a different drawing or image. Individual layers may be set to a desired layer opacity from the layers pane.

We can open as many windows into a map as we like. Each window is an independent view with its own pan and zoom.

**Creating New, Blank Components**

If at all possible, click open the map in which a new, blank component is to be used before creating that component. That will assure the new component is created using projection parameters that are a good match to the map.

The **File - Create** menu will add a "blank" component to the project. If a drawing, image, labels or map window is open when the **File - Create** menu is used, the projection of the window that is open will be used to create the new component.

The **File - Create** dialog takes the projection parameters for the new component from whatever window is active at the time the new component is created. If the project pane is active, the component will be created using the system default projection of Orthographic centered at the 0,0 world latitude/longitude origin. If a map window or other component window is active when the new component is created, the new component will be created using whatever projection is used by that active window. This context-sensitive setting of default projection parameters makes it much easier to create new components using projection parameters that are hassle-free by default.

When creating new, blank components to use as layers in maps it therefore makes sense to click open the map and to have it the active window when the **File - Create** command is invoked. This will automatically create the new component with projection parameters that match those in use in the map. This is a very important step to assure easy compatibility between the projections of newly created components and the map.

Another way to create blank components is with a right-click on a map's layer tabs and choosing **Add** to create a new component and add it as a layer to the map.

**Projections and Performance**
Maps re-project drawings and images on the fly into whatever projection is desired of the map. That can be a computationally intensive process when maps include many different images and drawings as layers and when the various drawings and images use different native projections. Maps can work much faster if the projection they use is the same as that used by the images and drawings they contain because then there is no need to re-project on the fly.

If we will be using drawings and images within the same map using the same projection it makes sense to re-project the drawings and images into the projection we use in the map. This is especially true if the map contains big images.

**Drawings and Images**

We can open any drawing, surface or image in its own window by double clicking on the component in the project pane. Drawings, surfaces and images opened in their own windows are shown using whatever native projection they use. If we wish to change the native projection used by a component we use the Edit - Change Projection dialog to change it. This is a permanent change in the structure of the data.

If desired we can use Manifold as a pure CAD editor or as a pure image editor (like PhotoShop) without worrying about any geographic contexts. When we import drawings or images from non-geographic formats such as .dxf or .jpeg they automatically are brought into Manifold as Orthographic projections georeferenced to the (0,0) world lat/lon origin.

For CAD work, we can combine CAD drawings with each other in multiple layers in a "map" just like AutoCAD allows use of multiple layers. For general-purpose image editing like Adobe PhotoShop we can combine images in layers in a map as well. If we switch the status bar read out from lat/lon to X.Y coordinate readout the "map" window will look just like any CAD editor's or photo editor's main workspace window. We'd never know that on a theoretical basis these drawings and images exist just off the coast of Africa or that technically they are in a projection. Remember, even at the thousandths of an inch resolution implied by 4800 DPI scanned images there is no difference over a few meters between an Orthographic projection and an unprojected coordinate system.

That every drawing and image are georegistered to some geographic default has no downside even if we never intend to use them in a geographic context. However, when it comes time to georegister an image or a CAD drawing (like a factory plan blueprint) it is very convenient that they are already georegistered to some default projection and location. Georegistering an image, for example, becomes a simple matter of moving it to the right place at the right scale and orientation and, if necessary, warping it using numerical transforms to match it to known-good geographic data.

Images may be grayscale, palette, RGB, RGBA or compressed images and may be freely converted between the various image types using the Image - Convert To command. An image must be an RGBA image to use per-pixel transparency. See the Image Types topic.

Large images intended for background use should always be compressed images for speed. For example, Manifold can open and display an ECW image that is many gigabytes in size virtually instantly.

**Selection**

Manifold has very rich, regularly organized ways of doing selections that are employed throughout the system. These selection methods must be mastered. Please do not attempt work with Manifold without thorough study of the Selection topic.

**Formatting**

Formats in drawings may be manually specified or automatically derived from the contents of data fields using thematic formatting. Use the Format toolbar to manually set formatting. Lines, areas and points in drawings may be formatted using foreground and background color, thickness or size, and style. All objects of a given type (line, area or point) in a drawing will all be formatted the same way. Text labels in labels components will also have the same format used throughout the labels component. To employ different formats for objects in a map, use multiple drawings.

Thematic formats may be set for foreground color or background color for points, lines or areas. To do so, click into the color well and choose Theme. See Thematic Formatting. Thematic formats to automatically color maps so no two adjacent regions share the same color may automatically be set using the Drawing - Color menu command.
The format toolbar is also used to set many of the display characteristics of elements added to print layouts. For example, we choose the font for a text box in a layout by first selecting it for editing (using CTRL-ALT click) and then making our choices in the format toolbar.

**Themes** are project components that store formatting choices for drawings. Opening a theme shows the drawing with the formatting specified by the theme. A drawing can have many themes, each of which may show that drawing through a different set of formatting choices. Adding a new theme to a project requires virtually no storage space.

**Labels**

Text labels are hosted in Labels components. Multiple labels components are stacked in map layers to create the label effects desired. Each labels component is formatted using the same format throughout the entire layer. To create labels that automatically show field information, we **Copy** a drawing and then **Paste As** a labels component to create a labels component that shows labels at the position of every object in the drawing. If we want to create labels for only some of the objects in the drawing, we first open the drawing and select some of the objects and then **Copy** them. The subsequent **Paste As** will paste labels only for the copied objects. See the Labels topic for more information.

**Georegistration**

We georegister images and CAD drawings using the View - Panes - Control Points pane. We can georegister an image to a drawing, a map or to another image. We can georegister drawings to images, drawings or maps. The usual task is to georegister a target image to a reference drawing using features that are visible in both as control points.

We can also georeference an image without needing a pre-existing drawing if we know the latitude / longitude coordinates of at least two features in the image. For example, we may have an image of a scanned paper map where we have used a GPS device to determine the exact coordinates of a few features in the map. We can use these coordinates to georegister the image so that it can be traced into a drawing.

See the Georegistration topic for details.

**Layer Restrictions**

When working with maps, individual layers in maps can be specified as selectable, editable or clickable using layer restrictions. This is an extremely important capability when creating sophisticated web sites where we do not want background layers to respond to clicks for tools such as the **Info** tool, and is very useful for interactive work in Manifold when we do not want to accidentally select or edit items in a background layer. See the Layers and Commands topic for more information.

**Vectorization**

Creating vector drawings from raster images is an interactive process within Manifold. Vectorizing is called **tracing** within Manifold. We use this term because everyone understands the idea of laying a sheet of tracing paper onto a photograph and then drawing lines with a pencil to outline what is seen in the photo. This is a much more accessible word for new users than "vectorization."

When tracing images to create objects in drawings there are several key controls we leave open for use:

- The View - Panes - Tools Properties pane is used to set tool properties such as sensitivity threshold for **Snap** and **Touch Select** and other tools we will use.
- The View - Panes - Layers pane will likely be left open to allow us to move quickly between layers in the map.
- The Tracing toolbar contains tools specific to vectorizing.
- The Tools toolbar contains editing tools for drawings and images. Drawing tools are used to create areas, lines and points. Image tools are used to interactively edit the colors of pixels. When a drawing window or layer in a map is active the drawing tools will automatically be created.
The speed, ease and quality of tracing depend almost entirely on the characteristics of the raster image being traced into a vector drawing. A very, very few raster images may be automatically traced with satisfactory results. Most images have so much pixel junk that fully-automatic tracing creates such unsatisfactory vector objects that it would have been quicker to create objects using heads up digitization than to correct all the errors in the automatic process.

Manifold has numerous tools with sophisticated tool properties that are designed to increase the speed and accuracy of heads-up tracing. All tracing happens within a map that includes at least one raster layer being traced and an active drawing layer in which objects are created. A typical scenario is to use several different drawing layers (say, with a layer for lines and a different area for points) and clicking each when we want the new objects to be created in that layer.

Most tracing is done using free-hand addition of lines and points into a drawing layer while using an image layer as a guide. If desired, automated tools may be used. The main automated tracing tools are:

- **Trace Point** - Point and click and automatically create a point at the center of a pixel cloud.
- **Trace Line** - Point and click to automatically create a line long the center of line-shaped pixel cloud.
- **Insert Point** - Create a point freehand at a given location.
- **Insert Line** - Create a line freehand at a given location.

Manifold includes a **Trace Area** tool to point and click to create an area within pixel regions. Depending on the quality of the raster image it may be faster to simply insert points and lines using the above tools and then create areas using the lines. See the Tracing topic for details.

**Surfaces and Terrains**

**Surfaces** are 2D visualizations of terrain elevation data or other abstract data. Terrain elevation data may be imported into Manifold from a variety of formats. Surfaces are always imported together with an accompanying **Terrain** component, which shows the data in a 3D view. We can create additional terrains for each surface (at no cost in storage) that show the same surface colored using different thematic palettes.

Surfaces may be viewed in shaded relief or otherwise used like 2D images and can be dragged and dropped as layers into maps. They may also be copied and pasted as images. The appearance of the surface is governed by the View - Display Options settings for the surface. Some of these options provide analytics. For example, a surface can be seen using slope or aspect calculations.

Terrains may be seen in 3D by double clicking open the terrain. When seen in 3D, a wide variety of options can be applied to the appearance of the terrain. Options include automatic coloring by value or overlay in high resolution with images or drawings.

See the Surfaces and Terrains topic for details.

**Tables and Database Management**

Manifold has absolutely **immense** database capabilities that dwarf any other general purpose GIS. Database operations begin by importing or creating tables. When importing a drawing one or more tables will automatically be created in association with that drawing.

Manifold can import tables from almost any source. When a table is imported into Manifold both the table and the data it contains are brought into the .map project file and maintained by Manifold. Take time to study the numerous capabilities of Manifold tables. The context menus available when right clicking onto columns, rows, and cells provide a wealth of services, such as more like this searches. Table menus also include important capabilities such as **filters, grouping** and **sorting** which may be multilevel.

Tables may also be linked into a project. Linked tables stay outside the .map file within their original database files or database provider. If the file format or provider is one supporting multi-user accesses, Manifold can open those tables and use them even as other applications may update them.

Manifold has three methods for reaching out to either import or to link to a table. Using **File - Import - Table** may be used to open any of the basic database file formats commonly used within Microsoft Office installations.
For more extended access to database files or providers, using **File - Import - Table** and choose **Data Sources** () in the **Files of Type** box to launch the Data Source dialog. The Data Source dialog allows us use whatever connection type is desired, for example, files of type **OLE DB, ODBC, ADO .NET** or native connection technologies such as **OCI**, the Oracle Call Interface. By default, Manifold installs the standard suite of Microsoft OLE DB providers as well as ADO .NET connections.

Manifold provides two specialized dialogs to manage interactions with external databases. The Database Console is available in all Manifold editions and can be used to connect to databases, to manage them, to execute queries within the DBMS server's own SQL and to rapidly import or link components from the database. The Administrator Console is available in Database Administrator Edition and provides specialized functions of interest to database administrators in settings where many people are using Manifold to connect to spatial DBMS servers.

**Relations**

Any table in a project may be linked to another table via a relation so columns from the one table become visible in the other table. Open a table in table window and then choose **Table - Relations** from the menu. Choose the table that is to be linked and the key fields in each table that will be matched to form the relation. Choose the fields from the linked table that are to be included when the table is displayed.

Relations in Manifold are like forming a relation in classic relational databases except that the relation formed is a "soft link" by default. In a classic relation, deleting a record in one table will also delete it in the table linked by a relation. When tables are linked in Manifold using the **Tables - Relations** dialog deleting a record in one table will not delete the linked records in other tables by default.

When a drawing is imported into or created within a project a table for it is created automatically. The table will contain at least one field, the object **ID** field, that is a key field connecting each object in the drawing to a row in the table. If drawings are imported with additional data fields for each object, those fields will appear in the table as well. Imports from some formats, such as SDTS, may also create additional tables that are joined by relations to the drawing's table.

Objects in drawings may also be connected to records in external tables by using relations. The process is simple. First, import the drawing. This will create both a drawing and the drawing's table. Next, link the external table into the project using **File - Link - Table**. Finally, open the drawing's table and use **Table - Relations** to form a relation between the drawing's table and the external table using some field in each to match up records.

**Transfer Rules**

Many editing / transformation operations in drawings (see Transform Toolbar - Drawings and Drawings - Dissolve, for example) will create new objects from existing objects. When new objects are created from existing ones the system must know how to transfer data attribute fields from the existing objects into the new objects. The Transfer Rules dialog in tables sets forth the rules to be used for each particular column.

**Queries**

Manifold includes full SQL / DDL / DML language capability. Create a query using **File - Create - Query** and then double click on it to open it for editing. Henceforth, when the query is run by right clicking on it and choosing **Run** it will execute the SQL statement and display the results as a table. Action type SQL queries may also be created in this way. Queries may be opened for editing and **Run** at the same time so that the SQL query and the table it produces may be seen at the same time.

Manifold SQL provides numerous extensions. See the Queries topic for more information on queries and the Spatial Extensions and Geocoding Extensions topics for a list of extensions to SQL in Manifold that allow creation of SQL queries that make spatial comparisons or employ the geocoding engine for work with street addresses. The **SQL Reference Guide** chapter in Help provides many detailed topics as well.

**Decision Support System**

The Decision Support System is a means of ranking records in tables by how they score in flexible queries. Flexible queries use flexible profiles as criteria instead of fixed criteria as used in traditional SQL.
Manifold provides two main charting systems:

- **Minicharts** are used with drawings to portray a small chartlet for each object in the drawing. These should be used with restraint to avoid overwhelming the viewer.
- **Charts** are created from tables. They show the contents of tables with multiple series (fields) grouped as desired.

### Analytics

Manifold has many methods of analysis.

- Analyses that create numbers are often done using **ViewBots**, which are one-line analytic instruments that can be set to look at the contents of a drawing or other component. ViewBots are a spectacularly useful part of Manifold.
- Analyses in tables may also be done using **Active Columns**, where the column dynamically reports the result of a formula.
- **SQL** may also be used for analyses, especially when using computational queries or crosstab queries to explore grouping.
- Analysis involving the creation of new objects, such as drawing a Voronoi tiling, will normally be accomplished using Transform toolbar operators. The transform toolbar is also used with tables to perform computations on fields and to manipulate the table.
- Transformation of images may be done with the transform toolbar as well as with **Image** menu tools.
- Some analytic operations are implied: for example creating a slope or aspect map from a surface is a simple matter of changing the display option for the surface to **Slope** or **Aspect**.

### Programming

There are many capabilities within Manifold that may be programmed in an "ad hoc" way. For example, **Active Columns** allow inclusion of an ActiveX program that computes the results seen in a table's column. For programming in the classical sense, Manifold approach to customization and programming takes one of two pathways:

- **Scripting** within Manifold is accomplished using Visual Basic Scripting Edition within Manifold's drag and drop scripting environment to create event-driven, forms-based scripts. Scripting using straight scripting text may be accomplished in any ActiveX scripting language if it is installed on the system. By default, Manifold uses VBScript and Javascript. If their scripting engines are installed on your system, you may also use other scripting languages such as PERL, Python or REXX. Cool!
- Manifold scripting can also be done using any .NET language, such as C#.
- Programming from external programs can use any Microsoft compatible language and development system, most likely Visual Basic .NET or Visual C++. Manifold itself is written entirely in Microsoft Visual C++ within the Visual Studio .NET development environment.

Manifold does not use VBA because we feel it is neither right for beginners nor suitable for experts. VBA is too clumsy for casual usage while being inadequate for professional programming. VBScript is a better choice for beginners and real VB or VB .NET or VC++ the right choice for professionals.

Our feeling is that industry-standard scripting languages have attained far greater support among computing users for casual usage. By using any ActiveX scripting language or .NET language for scripts in Manifold we provide a wide range of exquisite scripting languages to our users, all for free.

For professional program development, we feel that serious programmers already have invested in either VB or VC++ for program development. Using these languages, therefore, represents no additional cost for professional programmers while gaining the power and flexibility of a real compiled language and serious development environment.

Manifold also provides a Debugger similar to those used in professional programming environments for use with ActiveX scripting languages. The Debugger allows stepping through multiple scripts with automatic stops at breakpoints, paused execution and stepping into, over and out of routines. The **Call Stack**, **Variables** and **Watches** panes provide dynamic, editable displays of variables, functions, computed values and contexts.

See the Programming Manifold topic for more information.
Printing

Any component may be printed by opening it in a window and choosing File - Print. For more sophisticated printing composition, a layout component is created and opened and any other components to be printed are dragged and dropped into that layout. See the Printing topic.

Print layouts are highly WYSIWYG (What You See Is What You Get) so there is no need for a print preview.

Internet Map Server

Manifold Professional Edition and high versions include a powerful Internet Map Server (IMS) that can be used to create highly sophisticated map-enabled web sites. Many classes of web sites can be created with no programming required. Manifold IMS works on Microsoft Internet Information Server (IIS) machines.

To create a web site using Manifold IMS, create a Manifold project containing the component (normally a map) that you want to be published. Open the map and choose File - Export - Web Page. Choose the options desired and press OK. Manifold will automatically create the necessary .asp files required for the web site.

Although the GIS part of creating a web site with Manifold IMS is very simple, the Microsoft IIS side of operating web sites must be understood by the webmaster.

Multi-User Editing

Manifold supports full, concurrent multi-user editing of linked drawings with Enterprise Edition being required for full resolution of editing conflicts. See the Multi-User Editing of Linked Drawings topic for details.

Learning Manifold

The best path is to begin at the Read Me First topic and to then go through the Introduction book. Do not skip the Windows topic or other topics in the Introduction.

We suggest working through all of the Examples in sequence. As time permits, read through the topics in the Drawings, Images, Maps, Labels, Tables, Surfaces and Terrains and Queries books. It is OK to skip the detailed reference topics on various transform operators during a first reading.

Performance

Like all large, sophisticated professional tools Manifold provides great power and flexibility to users in how they choose to do GIS. There are often many different ways of accomplishing the same end in Manifold through different means. Users can choose the method that is exactly right for their intended goal and for their preferred means of reaching that goal. Providing such great flexibility and free choice for users also means that users can choose to conduct their GIS affairs in a highly inefficient way. For a guide to best practices for maximum efficiency, see the Performance Tips topic.

Notes

The reasoning behind Manifold’s user interface is explained in the essay titled User Interface Design. Other Essays present a sometimes radical Manifold spin on various topics in GIS that may amuse or irritate the expert reader.
Data Storage Strategies

How we decide to store our data says a lot about how we can use that data. Data in the case of Manifold consists of drawings, images, surfaces, tables and the like. Manifold provides such a rich spectrum of capabilities for storing and working with GIS data that sometimes beginners are confused by the many options possible.

The usual approach in the GIS industry (or, for that matter, in most other software industries) is to have two classes of storage, which may be loosely categorized as:

- **Local, Desktop storage** - Applications store data in individual files, just like Microsoft Word saves documents in .doc files or Excel saves spreadsheets in .xls files. This is simple and easy for individuals but not a good model for enterprises or multiuser applications.

- **Linked, Server storage** - Applications store data in centralized DBMS servers, to which users running client software can connect to link to whatever data they need. Enterprise-class database systems are usually used as the centralized server. This provides administrative centralization and support for multi-user work, but at the cost of added complexity.

Manifold can store data using either of the above models, and in addition introduces a third model, unique to Manifold, that splits the difference between the above two models:

- **Shared, Enterprise Edition** - This is a simplified Enterprise server shared storage model designed for use by small and mid-sized organizations which desire the simplicity of desktop storage while having some of the organizational advantages of server storage. Data is stored as components within the Enterprise server, like documents within a file cabinet, which can then be shared in their entirety into a particular project.

The result is three levels of storage supported by Manifold, which may be freely intermingled with each other. These three levels of storage may be summarized as:

- **Local components stored in .map project files** - This is the classic document-oriented desktop storage model: Users work with individual .map project files that contain the drawings, images, tables and other components they are working with. This is a typical approach for individuals or very small organizations, the classic "one file, one user" approach employed by Microsoft Office applications. Data is stored in the form of .map project files, which are the "documents," and the "file cabinet" used to store those documents is simply the Windows file system, using folders and other Windows facilities to keep things organized. There is no real multi-user activity done with this system. If someone else needs a drawing we've created, we give them a copy of the .map file containing the drawing. We hope that if they make any changes we will be able to reconcile any such changes with whatever other copies of that .map file we have on hand. Any Manifold version can use this storage model.

- **Linked components stored in database servers** - This is the classic, object-oriented server storage model. Drawings, images and tables are stored in a general-purpose way within a database server. Users can link such items into their local projects. This is the general-purpose, "one server, many users" approach used by large enterprises. It is called object oriented because the technology of linking drawings from general purpose geometry storage in database servers allows the entire drawing to be freely edited, down to creating, editing and deleting individual objects, that is, individual points, lines and areas. The "file cabinet" is the database server, and users can link entire components (drawings, images, tables, queries or surfaces) into their projects or, in the case of drawings if certain spatially-enabled DBMS servers are used, only that part of the drawing that is of interest. Concurrent, multi-user editing of drawings is fully supported. Enterprise Edition is required for full capabilities with this model.

- **Shared components stored in Enterprise servers** - This is a uniquely Manifold Enterprise server storage model that combines benefits of desktop and server storage models. Drawings, images, tables and other components are all kept in a centralized Enterprise server, which is simply a database configured in a Manifold-specific way to store all Manifold components. Manifold users running Enterprise Edition can link items from the Enterprise server into their project. This is a simplified "one server, many users" approach that provides component-oriented, server storage for small or mid-sized organizations. Individual users still can keep their projects saved as .map project file "documents," but the content of those documents is now fetched as whole components that are linked in from the Enterprise server "file cabinet." Multi-user editing does not occur simultaneously, but a shared component can be checked out by a user, edited, and then checked back into the server. Any changes made will be automatically propagated into any project that uses that shared component. Enterprise servers require use of Enterprise Edition on client desktops.
The above summary is just one way of looking at the many capabilities that Manifold provides for storing data. It helps explain why it is that Enterprise Edition includes a dedicated system for working with Enterprise servers even as it also includes very general-purpose features for saving linked drawings in DBMS servers using formats like OGC WKB or Oracle spatial SDO_GEOMETRY.

The advantage of Enterprise server storage is that it provides an easy-to-use way of storing any Manifold component, even scripts and comments and layouts, within a centralized server from which the component can be shared by many users within many projects. Enterprise servers provide simplified administration and ease of use for users. The check out / check in editing model provides easy editing by those who have permission to do so while removing the need for training users in the nuances of concurrent multi-user editing. The disadvantage is that Enterprise server storage is specific to Manifold only (although data can always be exported to other interchange formats) and does not allow concurrent multi-user editing.

Although it is more complex than using Enterprise servers, the general-purpose server storage model has a lot of appeal for interoperability with other programs. If a standard format, such as Oracle, is used, then potentially very many different applications can interact with that same geospatial data in a (hopefully) vendor-neutral way. The disadvantage is that only some types of components can be stored in this way and that greater expertise is required to exploit such servers. The availability of concurrent, multi-user editing is both an advantage and a disadvantage. Along with the obvious benefits comes the greater user expertise required to resolve editing conflicts that may arise from simultaneous editing of objects by different users.

One other potential use of server storage is the use of spatially-enabled DBMS servers to store GIS data. Such servers have the ability to perform some spatial operations, such as fetching all objects in a drawing within a particular area of interest, and therefore can be used to increase performance and capacity by offloading some work from the GIS desktop. See the Spatial DBMS topic for more information about spatially-enabled DBMS servers.

Manifold projects can mix and match the storage models above, and the intermingling of different storage models has been encouraged by advances in storage technology coupled with reductions in cost. In earlier days, for example, an individual user simply could not afford access to, say, Oracle spatial servers. In modern times, Oracle Express Edition can be used for free. Manifold also provides generic spatial DBMS capabilities for almost any DBMS, including free servers such as MySQL, which can now not only store drawings but also images and servers with almost unlimited capacity.

Because the cost of enterprise-class DBMS products has come down to zero in some cases, even individual users can employ server storage if that makes sense for them. Likewise, some aspects of server storage have appeared in the form of web servers that play a role (albeit at usually slower performance) once played exclusively by database servers. Users can freely combine storage models to achieve exactly the result desired.

For example, we can have a project that contains purely local components, links in an image from a web-based image server, shared queries or scripts from a shared Enterprise server and drawings linked in from an Oracle Spatial server. The linked image may be used for background only, so it doesn't matter that it cannot be edited, the scripts or queries may be something a more experienced user in our organization has shared to our department's Enterprise server, and the drawing might be an area of interest linked in from some vast, 200 GB "seamless" drawing maintained on our organization's Oracle cluster. Our job might be making edits to the vast Oracle drawing assisted by the scripts and queries provided by our colleague while the background drawing helps us stay oriented.

Advice to Users

The following guidelines reflect how most users tend to choose Manifold System versions and the storage strategy employed. Feel free to modify these guidelines for your own needs, as most users do.

- Individual users with relatively small and mid-sized projects usually keep everything in the same .map file, especially while learning Manifold. The main exception is that large images used for backgrounds should be saved in .ecw form and linked into projects. As users get more sophisticated and find themselves with very large data holdings, they will be careful to keep their Windows systems well organized.

- More advanced users with larger data holdings will often turn to Enterprise Edition to organize their data in Enterprise Servers. That's an especially useful strategy if a particular drawing is used in many projects (that could be stored on various machines in one's office) and the drawing is updated from time to time. It's a lot easier to simply check out the drawing, edit it and check it back into the Enterprise server than it is to track down each project that uses that drawing and then to manually update each project by copying and pasting from a reference project somewhere.
Most organizations with more than one person doing GIS will use Enterprise Edition, initially to store data within Enterprise Servers to help keep things organized. This is a good solution when simultaneous, multi-user editing of drawings is not required.

When simultaneous, multi-user editing of drawings is required, organizations will use Enterprise Edition and will save their drawings in a database such as Oracle, SQL Server, IBM DB2, PostgreSQL, MySQL or some other convenient DBMS.

Very large organizations tend to centralize their operations around storage in databases using either Oracle spatial technology for drawings or WKB geometry storage in SQL Server to allow interoperability with the many applications interacting with GIS data in such organizations.

Organizations working with many users and large images will almost always store the images in a spatial DBMS, either using Manifold's generic spatial DBMS capability with whatever DBMS is in use or choosing a spatial DBMS that includes raster support, such as Oracle Spatial with GeoRaster.

On occasion, individual users with large images or image libraries will also store them in a spatial DBMS, either using Manifold's generic spatial DBMS capabilities or using the free download of Oracle Enterprise Edition to get GeoRaster storage capability if their uses fit within the permitted scope of the free download Oracle license.

At any time, of course, if an application requires that a linked drawing be created from data stored outside the project (be it a personal database of some kind in Access .mdb format or be it a huge corporate database leveraged by a big IMS application), then we will use linked drawings or linked tables as necessary to bring data into our project from other data sources.

Note that at any time we can always convert a linked or shared component into a local component by unlinking it or unsharing it. Doing so makes a snapshot of the data supplied by the data source as of that instant. This is a handy way of capturing entirely within the project's .map file the current content of a project that includes linked components, so that we can send that snapshot to a colleague who might not have the ability to link to the same data sources.
Spatial DBMS

This topic continues the discussion begun in the Data Storage Strategies topic. In this topic we look more closely at storing GIS data, in particular drawings, within database management system (DBMS) packages. This topic provides a general introduction to the Manifold approach to spatial DBMS. See the Spatial DBMS Facilities topic for detailed information on Manifold spatial DBMS capabilities, including information on supported spatial DBMS products.

The use of spatial DBMS technology is a great way to store and work with drawings and images of immense size with high performance. Manifold provides a wide range of capabilities to take advantage of spatial DBMS:

- Manifold can store drawings within major spatial DBMS products (IBM DB2 with IBM Spatial Extender, Microsoft SQL Server 2008 spatial, Oracle and PostgreSQL/PostGIS) using the native spatial DBMS facilities within those products.
- Manifold can store drawings, images and surfaces in almost any DBMS using Manifold-managed spatial DBMS capabilities.
- Manifold can store images and surfaces within Oracle DBMS products supporting Oracle native GeoRaster capability.
- In addition to using Manifold or native spatial DBMS capabilities, Manifold can also connect to ESRI-style SDE geodatabases and Personal geodatabases and import and link (for read/write editing) drawings in SDE or Personal geodatabases.

DBMS products have become so powerful, so inexpensive and so easy to install that more and more GIS applications now take advantage of DBMS. GIS users from individual hobbyists to the largest enterprises can now benefit from spatial DBMS technology through Manifold.

Manifold Enterprise Edition or above is required to connect using native connections to spatial DBMS such as Oracle Spatial, DB2 with IBM Spatial Extender, SQL Server 2008 spatial and PostgreSQL / PostGIS. While Personal and Professional editions can make use of spatial DBMS in some cases once configuration has been done with Enterprise Edition, in general Enterprise Edition or above is required for spatial DBMS work.

Two Types of Data in Drawings

Most GIS packages, Manifold included, use two types of data to make up drawings:

- One type of data is the geometric information that defines objects in the drawing. This type of geometric information is often simply called the geometry of the drawing. It specifies the shape and location of the points, lines and areas of which the drawing consists.
- The second type of information is the data in the optional attributes that may be attached to objects in a drawing. For example, a drawing that consists of points showing the locations of cities might also have attributes that give the name of each city, the population of that city and other information.

GIS packages use geometry to draw the shapes and locations of points, lines and areas in a window. Attribute data is normally shown using a row and column table visual interface like that used in DBMS packages. A drawing may or may not have attributes for the objects it contains, but it always has geometry because without geometry there would be no points, lines or areas in the drawing.

Storing Attribute Data in a DBMS

A GIS package becomes more flexible, useful and powerful if it can work with attribute data that is stored in an external DBMS. That's very useful even if the GIS continues to manage all of the geometry.

Many GIS packages have this capability, although it is usually limited to read-only linkages or available only in a limited number of data access methods, such as ODBC. Manifold allows linking of attribute data from an external DBMS with full read / write / edit capabilities using a very wide range of data access methods. See the Relations topic and the Attaching External Tables to Drawings topic for Manifold techniques for using attribute data from external sources.
Linking attribute data from a DBMS brings a lot of advantages to GIS users. The foremost advantage is that there is much more data maintained in commercial DBMS products than exists in GIS storage, so being able to link data into GIS projects from an external DBMS gives access to a much wider range of data.

The second key advantage is the ability to dynamically update a GIS project using data that is maintained in some external DBMS. Suppose, for example, we want to graphically display the sales of our company in various regions by thematically changing the color of regions in accordance with a sales attribute for each region. If the sales attribute is automatically fetched into the drawing from our corporate DBMS, whenever that corporate DBMS is updated by some other process (such as, perhaps, by our company’s order entry system), our drawing will automatically be updated as well.

A third advantage is that storing attribute data in a DBMS also can take advantage of ubiquitous data interchange between applications that can work with DBMS. If our attribute data is stored in a DBMS there are likely thousands of applications and utilities that can work with it without having to know anything about our GIS package. Applications can connect to the DBMS as they have long known how to do, while the GIS takes data as needed from the DBMS.

**Storing Geometry Data in a DBMS**

There are many benefits to storing attribute data in an external DBMS even though geometry continues to be managed exclusively within the GIS. But so long as the GIS manages geometry users will be subject to whatever limitations that may involve. There are usually several limitations imposed by having geometry managed exclusively by the GIS package.

The first limitation is that GIS packages usually do not have the sheer capacity and cluster scalability of mainstream DBMS packages like Oracle or SQL Server. As a result, the number of objects in a drawing will normally be limited by the performance of the GIS running on a single machine.

A second limitation is often a restriction to single users or single processes working with the geometry data stored by the GIS. Mainstream DBMS packages have evolved to meet intensely multiuser, multi-process needs but most GIS packages (Manifold is an exception) have not been built with the multi-process, transaction-oriented architecture required for intensively multiuser operations on geometry.

A third limitation is limited dynamic interoperability between different GIS packages and other applications. GIS data can be interchanged more or less successfully in a static, limited way using well known formats such as shapefiles, but cannot in general be interchanged dynamically as is taken for granted with DBMS servers.

A GIS that can store geometry in a DBMS can escape the above limitations. Storing geometry in a DBMS can take advantage of the capacity and scalability of the DBMS, resulting in drawings that can be terabytes in size. If a DBMS stores geometry, a GIS can operate as a client to the DBMS server in multi-user settings, taking advantage of the extensive apparatus a modern DBMS will provide to manage the simultaneous use of data by many different users. Storing geometry within the DBMS using data types native to the DBMS allows interchange with any application that understands those data types, which is usually many more applications than can parse the proprietary internal formats used for geometry storage within even very popular GIS products.

Manifold can work with geometry stored in databases using the widest and most flexible range of data access methods of any GIS system. In fact, Manifold can do so simultaneously with many different data sources and access methods, including geometry stored within Manifold itself.

**Native Geometry and Non-Native Geometry Types**

All data stored in a DBMS is ultimately in binary form. When storing geometry within a DBMS the question is what internal format the DBMS should use to order the binary data used to store that geometry. There are two approaches.

One approach is used in DBMS products sold as “spatial” DBMS. A spatial DBMS will have a pre-defined way of organizing binary data to represent geometry, and this pre-defined way of organizing binary data is built into the DBMS in the form of a data type, such as `SDO_GEOMETRY` in Oracle Spatial or `ST_GEOMETRY` in IBM’s DB2 Spatial Extender. Because this data type is built into the DBMS it is called a *native* geometry type. The data is still binary data, of course, but it has been organized in accordance with a formal expected for geometry data by the DBMS. When a DBMS product has its own, native geometry type it also usually supports that data type with additional infrastructure, such as the automatic creation of spatial indices or the provision of DBMS server commands that understand that data type.
The other approach is used with DBMS products that do not specify a pre-defined way of organizing binary data to represent geometry but which allow applications to utilize a generic binary data type. Almost all modern DBMS packages provide a generic binary data type that can be used to store binary data unstructured by the DBMS. Often referred to as a blob, such generic binary storage can be employed by applications as they see fit. When such generic binary storage is used to store geometry in a form not built into the DBMS it is called a non-native geometry type. Although the use of non-native geometry types allows storing geometry within general-purpose DBMS products without requiring a special "spatial" form of a DBMS, it does require a GIS application that supports the geometry formats to be used.

Manifold can use either native or non-native types of storage when available within a DBMS and can even use both types simultaneously. It is important to note that in either case the data is stored in binary form using some organizing format to represent geometry in that binary data. The difference is that in the one case the DBMS has its own special format for organizing that binary data and in the other case it is the GIS interpreting that binary data. Both cases can be equally fast if the non-native type is an efficient geometry format such as Manifold's SDO_GEOMETRY, since in both cases the speed is a function of the DBMS's ability to work with binary data, perhaps as assisted with specialized indices.

The main advantages of using a native geometry type within a spatial DBMS is first, that doing so provides interoperability with any application that uses the native type, and second, using a native type automatically takes advantage of the infrastructure within the spatial DBMS that supports that native data type. The disadvantage is that a native geometry type might not be available within the DBMS package we desire to use, or that it may involve extra cost.

The main advantages of non-native binary storage are first, Manifold makes use of spatial DBMS functionality utilizing non-native types within virtually every DBMS and second, using a choice of non-native geometry types can open the door to greater flexibility than is possible by committing to a single data type within a single DBMS vendor. A possible disadvantage is that choosing from a variety of geometry types can make interoperability with other applications more difficult. However, if a generic geometry type that is well-understood and accepted by many applications, such as WKB, is used, then interoperability might well be preserved.

**Spatial DBMS**

A spatial DBMS can make it especially convenient to leverage the power of the DBMS by using native geometry types within the DBMS and by extending the internal logic of the DBMS to handle spatial operations of interest for GIS applications. When a DBMS offers a native geometry type together with supporting capabilities (such as spatial indices) it is referred to as a spatial DBMS.

GIS has become such big business that major DBMS vendors have extended their DBMS products with native geometry types as well as with supporting capabilities. Locator capability within regular Oracle DBMS editions as well as Oracle's dedicated spatial product, Oracle Spatial, provide Oracle's SDO_GEOMETRY type as well as Oracle GeoRasters and powerful spatial operations. Other DBMS packages such as IBM DB2 have spatial extenders and open source DBMS packages such as PostgreSQL now include "spatial" capabilities as well.

Microsoft SQL Server allows third parties to extend the DBMS product to provide a native geometry type and spatial operators, as several third parties have done. In addition, Microsoft's new SQL Server 2008 includes Microsoft-engineered geometry types, GEOMETRY and GEOGRAPHY, together with supporting spatial capabilities.

The availability of a spatial DBMS with a native geometry type together with built-in spatial operators confers enormous benefits to a GIS package that can integrate well with that spatial DBMS. Capacity and scalability can increase to whatever the DBMS can handle, terabytes in the case of modern DBMS server clusters. Even if a GIS cannot manage terabytes at one time, by using the spatial capabilities of the DBMS to extract data in manageable area of interest (AOI) subsets, a GIS user can work effectively with that part of the data of interest taken from a data set that is much larger than the GIS client alone could handle.

When geometry is stored in a spatial DBMS multiuser and transaction issues need no longer be limited by the GIS but instead can be handled by the formidable apparatus of the DBMS while the data itself can be protected by the transactional integrity of the DBMS. DBMS servers provide transactions, triggers, views and many other capabilities not often found within a GIS.

Because mainstream DBMS packages are used by organizations for many more applications than GIS, storing data within the DBMS package's own native data types tends to assure interoperability with a far greater number of applications than can work with any one GIS vendor's proprietary formats, potentially including a greater number of GIS packages as well. GIS vendors may not be able to agree among themselves to open their formats to each other, but every GIS vendor knows they are not in the game if they do not support Oracle's SDO_GEOMETRY or Microsoft's new GEOMETRY type in SQL Server 2008 spatial.
Spatial Indices

The spatial logic in a spatial DBMS normally consists of enabling infrastructure in the form of spatial indices maintained by the DBMS together with a collection of spatial operators that can be executed server-side by the DBMS server. A spatial index is a specialized form of DBMS index maintained for the geometry type that makes it possible to quickly identify objects based upon spatial characteristics, such as the location and extent of a particular object. Spatial operators are normally functions that ascertain spatial relationships, such as finding all objects that are contained within a given object.

When Manifold adds spatial capability to an ordinary DBMS, spatial indices will be created and used. There are important requirements and limitations to spatial indices in such cases that are covered in the Manifold Spatial DBMS Facilities topic.

Adding Spatial DBMS Capability to "Ordinary" DBMS

Manifold can confer spatial DBMS functionality to almost any DBMS package even if the DBMS does not itself have "spatial" features such as a native geometry type or spatial indices. Manifold does this by providing Manifold-sponsored spatial infrastructure within the DBMS to support usage of non-native geometry as is done with native geometry types within a spatial DBMS. Manifold utilizes geometry types desired by the user and creates spatial indices which the DBMS then operates.

Note that there is nothing about a "spatial" DBMS's native geometry type which makes it any better or worse as a binary format for storing data than, say, Manifold's own GEOMETRY type. Embed within a DBMS a spatial index tailored for GEOMETRY and we have almost all of what people really use within a spatial DBMS. In fact, Manifold GEOMETRY is actually significantly more efficient and powerful than the native geometry types used by either DB2 or PostgreSQL. It is just that those DBMS have some facilities that assume their native geometry type is in a particular format.

When Manifold provides spatial infrastructure, Manifold creates a spatial index in the target DBMS for Manifold geometry data stored in the DBMS. In such cases, Manifold stores geometry data as a binary blob, but it does so in a way the DBMS would do if it had its own geometry type, that is, with a spatial index maintained to allow sophisticated and powerful manipulation of that geometry data. The spatial index established by Manifold is maintained by the DBMS server using the power of the DBMS package and will automatically be updated if a linked drawing is edited. This confers upon ordinary DBMS packages a true "spatial" capability maintained together by the DBMS and by Manifold.

Manifold requires no add-ins to be installed in order to provide the above spatial capabilities to virtually any DBMS. However, because SQL Server 2005 is so popular in the Manifold user community and because SQL Server 2005 allows add-ins to extend functionality within SQL Server 2005, Manifold provides the free Manifold Spatial Extender for SQL Server 2005, an add-in that enables Manifold-managed spatial DBMS capability within SQL Server 2005 to operate with especially high performance.

Some people may say that a DBMS cannot be considered a true "spatial" DBMS unless in addition to having a geometry type plus supporting spatial infrastructure such as spatial indices it also has the ability to do server-side spatial operators, for example, to have operators such as a CONTAINS within the DBMS package's own SQL. Server-side operators are very important when the spatial DBMS is used as the "black box" infrastructure behind some application, such as a web site where data from the database is fetched using SQL queries or other code. However such server-side spatial operators are not usually employed when a rich GIS package is used as a "front end" to the spatial DBMS.

Server-side spatial operators are almost never used in an environment where the interaction with the spatial DBMS occurs through a very rich client like Manifold. The reason is that most people prefer a point-and-click user interface for spatial tasks and find it difficult to work within a black-box, command-line (that is, exclusively SQL) environment as often ends up being the only useful way to work with server-side spatial components. As a practical matter, when a GIS package like Manifold provides the user interface perhaps the only server-side operation executed by most users is an Area of Interest (AOI) specification to grab a manageable subset of some very large drawing. However, AOI does not require server-side spatial operators - it requires only a spatial index as conferred by Manifold.

Therefore, the use of a Manifold-declared spatial index together with straightforward Manifold GEOMETRY storage within an "ordinary" DBMS such as MySQL or Microsoft SQL Server 2005 (as opposed to built-in spatial support in SQL Server 2008) can provide the benefits sought by users through "spatial" DBMS. It is a way for GIS users to get outstanding price/performance by using ordinary DBMS packages, even possibly free packages, as true spatial DBMS servers storing data of effectively limitless capacity, storing drawings, images and surfaces, and to leveraging the power of increasingly multiprocessor systems and increasingly immense RAM.
In fact, by using the spatial DBMS capability conferred by Manifold onto a regular DBMS we can usually gain more "spatial" capabilities than are provided by some spatial DBMS packages. For example, Manifold conferring spatial capability onto an ordinary DBMS allows us to store images and surfaces as well as drawings while most spatial DBMS packages can only store drawings. See the Spatial DBMS Facilities topic for information on storing images and surfaces.

**Notes**

Users sometimes ask why Manifold supports so many different ways of doing spatial DBMS, at times even different ways using the very same DBMS product. The answer is that the Manifold user community includes a very large number of different organizations and individuals who have very diverse interests and the company likes to provide as many options as possible to suit different tastes.

Although it is true that providing "generic" spatial DBMS capabilities which will also work within a native spatial DBMS such as Oracle may seem redundant, the internal architecture of Manifold is so modular that there is not a great engineering cost for making all forms of spatial DBMS storage available regardless of what data store is employed. This provides flexibility and freedom of choice which helps users get the maximum possible value out of their technology investments.

For example, although using native Oracle Spatial capabilities is clearly the first choice for GIS professionals working with Oracle, it could be that to take advantage of some third party utility or application that does not understand native Oracle Spatial but does understand, say, OGC WKB we would like to make some data available in our Oracle data store in OGC WKB form but still have a generic spatial index in play for good performance. Manifold makes that possible and easy.

Although all Manifold editions can work with attribute data and can store geometry using blobs, only Enterprise Edition and above can take advantage of spatial DBMS to store geometry using a spatial DBMS server's native geometry type.

This topic has discussed two forms of data, geometry and attribute data, in connection with drawings. However, there is also a third class of data, raster data for images and surfaces, that is also often stored within a spatial DBMS. All spatial DBMS packages enable storage of geometry for drawings, but not all enable storage of raster data for images and surfaces.

Manifold can work with raster data in some spatial DBMS packages, for example, Oracle Spatial. However, even within a particular DBMS product line raster capability may be an option not found in all versions of that DBMS. For example, the free Oracle Express package has Locator capability in it to allow spatial DBMS work with drawings, but Oracle Express does not include Oracle's GeoRaster type to enable work with images. One must step up to the full Oracle Spatial package to get GeoRaster capability.

When using Manifold's generic spatial DBMS capabilities, nearly any DBMS can be used to store images and surfaces. Excellent!

**See Also**

Spatial DBMS Facilities - A key topic for anyone working with spatial DBMS.

Data Storage Strategies
Database Installations
Drawing - Area of Interest
ESRI Geodatabases
Example: Storing a Drawing in Manifold Spatial DBMS
Example: Storing an Image in Manifold Spatial DBMS
Example: Storing a Surface in Manifold Spatial DBMS
Geometry in Tables
Manifold Spatial DBMS Facilities
Manifold Spatial Extender for SQL Server
Linked Drawings
Oracle Spatial Facilities
Project Pane - Open Data Source
Queries and Geoms
Spatial Extensions
SQL Server Spatial DBMS Facilities
ESRI Geodatabases

This topic continues the general introduction to geospatial data storage covered by the Data Storage Strategies and the Spatial DBMS topics. This topic should be read together with those two topics, which introduce terminology and concepts used in this topic as well. This topic breaks out information specific to ESRI use of databases for new types of ESRI geospatial storage for the convenience of ESRI users. The features in this topic require Manifold Enterprise Edition or above.

ESRI, a legacy GIS vendor, has introduced a variety of products that can work with geometry and attribute data stored in DBMS servers. These are not spatial DBMS servers as such are generally understood, but rather are software products and middleware software that manage data stored in real DBMS servers using either blob or geometry types if the DBMS is a spatial DBMS. It is a classic case of a GIS vendor providing their own spatial capabilities for a DBMS by using their own non-native geometry types together with supporting capabilities supplied by the GIS vendor.

Manifold can also work with non-native data stored in databases using ESRI conventions. In such cases Manifold will work with ESRI's geometry types and supporting metadata tables using ESRI conventions for compatibility with ESRI products.

Nomenclature

Many users are baffled by ESRI nomenclature when it comes to parsing the bewildering variety of marketing phrases ESRI has used to describe ESRI "geodatabase" formats. If you feel baffled, you are not alone. In a nutshell, ESRI at one point introduced the idea of storing geometry in DBMS using a format that more or less boiled down to storing shapefiles within blobs. This was done in a complex way using middleware called ArcSDE that worked with serious databases like Oracle, and it was also done in a somewhat simplified way in Personal Geodatabase products that used Access .MDB files and were later apparently updated to work with MSDE (a free version of Microsoft SQL Server) or with SQL Server Express. In recent years, the ArcSDE product name seems to have been dropped by ESRI: more recent versions of this technology have been packaged as part of the ArcGIS product family and have been referred to as geodatabases.

All such storage methods are technically similar and are generally referred to as SDE geodatabase formats or as Personal geodatabase formats when in the somewhat simplified form that uses Access .MDB for file-based storage. Since all such formats are similar or derived from ArcSDE, they are referred to by Manifold documentation as ESRI SDE or as ESRI Geodatabase or as Personal Geodatabase data sources, the various terms being used interchangeably, regardless of which file format or DBMS system is used to store the data.

The terms are used interchangeably because some ESRI users come from a long ArcSDE tradition and don't realize that "geodatabase" is the new term for the same old thing, while some newer ESRI users might not realize that their "geodatabase" is really SDE technology with a new name. Because of the confusion caused by ESRI names for their SDE and their Personal technology being so similar, Manifold documentation will often refer to SDE and Personal geodatabases to underline that a particular capability is available whether one is working with either SDE geodatabases or the somewhat simpler Personal geodatabases.

Manifold can also connect to ESRI SDE and Personal geodatabase data sources for full read / write / edit capability. The only limitation is that unlike all other work with all other spatial DBMS, Manifold will not create new SDE or new Personal geodatabases, nor will Manifold add new drawings to an existing geodatabase. If we already have drawings in an SDE or Personal geodatabase, Manifold will happily import or link to those drawings. We can edit those drawings, adding new objects and deleting or editing old objects and in general perform whatever operation we like. For example, we could link to an existing drawing in a geodatabase and then copy and paste objects from some other drawing into that drawing. However, we cannot create new drawings or new geodatabases.

ESRI ArcSDE / ArcGIS / Personal Geodatabases

ESRI's ArcSDE product stores drawing geometry and other GIS data within ordinary, non-spatial DBMS servers. ESRI products refer to such data as geodatabases or SDE data sources (see notes on nomenclature above).

Technically, one can organize an SDE data source on almost any database. However, since this can not be done in a database-neutral fashion and since setting up an SDE data source involves creating database-specific objects, SDE data sources are only organized on big-name databases explicitly supported by ESRI, such as IBM DB2, IBM Informix, Microsoft SQL Server 2000 or 2005 and Oracle. SDE data sources using Access .mdb appear to have been replaced with SQL Server Express 2005. Personal geodatabases seem to be found almost exclusively within .mdb files.
Manifold knows to look for SDE or Personal geodatabase data if we use Database Console to connect to a database. Manifold usage of SDE or Personal geodatabase data sources uses Database Console as the primary interface and includes:

- Connecting to an SDE data source.
- Listing the drawings in an SDE data source in Database Console.
- Importing drawings.
- Linking drawings in read-write mode.

When importing or linking drawings from SDE or Personal geodatabase data sources Manifold will fetch the coordinate systems (projections) in use from ESRI metadata. Importing or linking a drawing assigns it the coordinate system stored on the data source.

Manifold will convert ESRI style objects within the SDE database into Manifold equivalents. For example, reading data from an SDE geodatabase reads parametric curves, flattening them into lines with straight line segments. As of the current writing Manifold does not accept "multipoint" values, although this capability is expected to be added in future editions.

Although Manifold can connect to an existing SDE database, read (import) drawings, write drawings, link drawings and edit drawings, Manifold will not export new drawings to an SDE database nor will Manifold create a new SDE database.

See the Spatial DBMS topic for more information on the advantages and disadvantages of spatial storage within a DBMS.

Notes

Why doesn't Manifold create new ESRI geodatabases?

There are two reasons for this limitation. The first reason is that it does not make sense to create new SDE databases since these are significantly inferior to using native standards within a true spatial DBMS like Oracle or SQL Server. This should not be an obstacle for ESRI users who are operating legacy SDE databases, since such users normally have ESRI software available should they ever need to create a new SDE database.

The limitation does not affect non-legacy users since new GIS users will never choose SDE over a real spatial DBMS. In modern times it does not make sense to create a new SDE database when true spatial DBMS servers, like Oracle Spatial or SQL Server 2008 spatial, are now available. Using a true spatial DBMS provides far greater reliability, performance, flexibility and interoperability with other applications and usually dramatically lower costs as well. It therefore makes no sense to use SDE if we have a real spatial DBMS available.

The purpose of Manifold's ability to work with SDE is therefore to provide the capability of dynamically working with existing SDE databases in legacy installations so that ESRI users can enjoy the convenience of working with that data and, eventually, rescue it when they transition to modern spatial DBMS.

The second reason is that from a technical perspective both SDE and Personal geodatabases are regarded by Manifold engineering as lacking desired reliability due to the convoluted architecture employed in those ESRI technologies. Simply put, the manifold.net team does not want to be blamed for problems arising from that lack of robustness. If someone wants to create a new drawing in an ESRI geodatabase and they run into problems when using an ESRI tool to create that drawing, then the problem is clearly ESRI's fault. In contrast, if Manifold were used to create that new drawing no matter how well Manifold accomplished the task if problems arose due to the basic architectural unreliability of ESRI geodatabases it would be Manifold that would be blamed. No thanks!

See Also

Data Storage Strategies
 Spatial DBMS
 Spatial DBMS Facilities
 SQL Server Spatial DBMS Facilities
The Data Source Dialog

GIS work often involves reading and writing information stored in database servers or data files. Manifold allows us to work with information in databases in many different ways and allows us to connect to all database and data file technologies in common use in Windows systems.

Examples of connecting to data include:

- Connect to a SQL Server database server within Database Console to examine the contents of that database.
- Upload an image from a Manifold project into an Oracle Spatial DBMS.
- Link a surface into a Manifold project from a MySQL DBMS.
- Link a drawing from a SQL Server database.
- Import a table from an Access .MDB file into a Manifold project.
- Link a table from an Access .MDB file into a Manifold project.
- Import a table from a .CSV text file.

Manifold provides two pathways to connect to data:

- For simple connections to simple data, we can use the File - Export, File - Import and File - Link commands to connect to simple file types such as CSV, DB, DBF, HTML, MDB, WK and XLS directly by choosing the desired file type in the Files of Type box in the dialogs. This is similar to how we would work with such files in typical Windows applications.
- For more complex data sources or more sophisticated uses, when launching the File - Export, File - Import and File - Link commands we would choose Data Sources () in the Files of Type box in the dialogs. This launches a standard Manifold dialog, the Data Source dialog, as a simple user interface for connecting to databases and data files. The Data Source dialog can also connect to simple file types as above and is used to connect to those simple file types for more sophisticated uses, such as linking a drawing from a table stored in a file. The Data Source dialog is always used when linking a drawing or linking a surface.

The Data Source dialog remembers a list of previously-used data sources and allows point and click connection without requiring users to remember what can be complex connection settings or obscure locations on hard disk.

The remembered list of previously-used data sources will be available within the Data Source dialog no matter what part of Manifold needs to connect to a data source. For example, it is frequently the case that if we import data from a database we might also want to export data to that same database. It is therefore very convenient to have that database connection already in the Data Source dialog's list whether we launch commands from the File - Import or the File - Export menus.

The first time we use a data source we will have to add it to the list in the Data Source dialog. Thereafter, the data source will be remembered in the list.

The Data Sources dialog works with all data source connection technologies supported by Manifold. The Data Sources dialog works with the Database Console, the Administrator Console and is also available when importing, linking and exporting components such as drawings, images and surfaces via the File - Export, File - Import, File - Link and similar dialogs. Choosing Data Sources () in the Files of Type box in such dialogs will invoke the Data Sources dialog.

Data Sources

There are generally two types of data sources we use with Manifold:

- **File-based data sources** - These are simply data files, such as .CSV (comma separated values) text files, .XLS (Excel), .DBF (dBase II / FoxBase), .MDB (Access) or other files. Some such data files can be very simple, such as plain, human-readable ASCII text files while other such data files, such as .MDB, can be more complex and require software that understands their internal organization to utilize. What file-based data sources have in common is that they all store data in files within Windows and applications that want to utilize the data in those files will open, read and write those files using some suitable connection methodology. Manifold can connect to all popular data file formats using either a connection technology specific to that data file such as, for example, Microsoft Jet for standard...
Microsoft file types such as Access .MDB or Excel .XLS, or a generic connection technology such as ODBC, OLE DB or ADO .NET.

• **Database servers** - DBMS servers are software products like Oracle, SQL Server, DB2, PostgreSQL, MySQL and similar that provide information through software connections. When we connect to a DBMS server we may not know exactly how the server stores information since we never actually connect to a specific file. Instead, the DBMS server provides whatever information we need. Manifold can connect to all popular DBMS servers using either a connection technology specific to that server such as, for example, OCI (Oracle Call Interface) for Oracle, or a generic connection technology such as ODBC, OLE DB or ADO .NET.

Either of the above types of data source can be added as a data source within the Data Source dialog. In addition, a third way of specifying a data source connection, DSN and UDL files, can also be specified as a data source.

DSN and UDL files do not themselves store data but instead are a way of saving within a file the information required to establish an ODBC or an OLE DB connection with a given data source, which could be file-based storage or a database server. A DSN connection describes a connection using ODBC and a UDL file describes a connection using OLE DB. Opening the DSN or UDL file within Manifold establishes the connection specified by that file.

**Using the Data Sources Dialog**

On first use the Data Sources dialog has no data sources listed. We first add a data source to the list. We can then double-click the data source to connect. The Data Sources dialog will remember data sources for future use, including in future Manifold sessions.

**To Add a Data Source**

1. Choose an operation, such as pressing the ... button in the Database Console, that launches the Data Sources dialog.
2. Click on the Add Data Source button. This launches the Connect To dialog.
3. In the Connect To dialog choose the connection method desired in the Files of type box and proceed through the dialogs for that connection method to add the data source.

**To Connect to a Data Source**

1. Launch the Data Sources dialog.
2. Double-click on a data source, or, click on a data source in the list to highlight it and press OK.

**To Delete a Data Source**

1. Launch the Data Sources dialog.
2. Click on a data source in the list to highlight it.
3. Click on the Delete button.

**Data Sources Dialog Commands**

- Add Data Source - Add a new data source.
- Delete - Delete highlighted data source.

(Contents pane) Displays a list of data sources, showing the name and type of connection for each.

Name Name to use for the highlighted data source.

(Connection string) Connection string to use for the highlighted data source.
Click to edit connection string using the standard dialogs for that connection technology.

Any PASSWORD and PWD parameters used in a connection string will be masked in the connection string to protect security.

Data Source Connections for Spatial DBMS

In addition to the usual roster of ADO .NET, OLE DB, and ODBC connections Manifold Enterprise Edition and higher editions provide specialized native DBMS connection choices that are used when adding a new data source using a spatial DBMS in the Data Source dialog.

DB2 Data Sources
Connect to IBM DB2 equipped with the DB2 Spatial Extender using IBM’s native spatial connection technology.

Oracle Data Sources
Connect to Oracle using OCI, the Oracle Call Interface to utilize Locator or Oracle Spatial facilities.

PostgreSQL Data Sources
Connect to PostgreSQL equipped with the PostGIS spatial extender using native PostgreSQL connection technology. Connections made using this data source will have any password required masked so that later usage will not inadvertently expose the password.

SQL Server Data Sources
Connect to Microsoft SQL Server 2008 spatial DBMS facilities using native SQL Server 2008 spatial connection technology.

Users sometimes ask if there is a special connection when using SQL Server 2005 as a spatial DBMS with the Manifold Spatial Extender for SQL Server installed. There is no need for a special connection because Manifold when connecting to SQL Server using any OLE DB connection will recognize and automatically use spatial indices managed by the spatial extender when the Manifold spatial extender has been installed.

Example: Create a Data Source for SQL Server Express

Let's add a connection to a local SQL Server Express data source to the Data Sources dialog. This is a straightforward connection for using SQL Server 2005 and not a native connection to SQL Server 2008 spatial.
We begin by launching Database Console and then clicking the [ ] button to specify a data source.

Click the **Add Data Source** button in the Data Source dialog.

In the **Connect To** dialog choose **OLE DB Data Sources ()** in the **Files of type** box.
In the Data Link Properties dialog’s Provider tab we choose the SQL Native Client and press Next.

In the Data Link Properties dialog:

1. Enter the data source and/or location of the data:
   - Data Source: \(\text{(local)\:SQLExpress}\)
   - Location:

2. Enter information to log on to the server:
   - Use Windows NT Integrated security
   - Use a specific user name and password:
     - User name:
     - Password:
     - Blank password: [ ]
     - Allow saving password: [ ]

3. Enter the initial catalog to use:

Pressing the Test Connection button is almost always a good idea at this point, to make sure there is nothing grossly wrong (such as a typographical error) in the information provided. When we are happy the connection is live, we press OK.

In the Connection tab we specify the settings used for our local SQL Server Express installation. In the case of this example we use the default data source name of \(\text{(local)\:SQLExpress}\) and Windows NT integrated security, which is the default recommended by Manifold’s SQL Server Express installation instructions.
Back in the Data Sources dialog we see that a new data source has been added using the default name for the connection. It is usually wise to enter some more memorable name that will help us recognize data sources in a more useful manner. This is easy to do.

In the Name box we can enter the friendly name we would like to use to identify this data source. Given the sophistication of data connection technologies available in Windows we might have many different data sources we use. For example, we might have a connection to a DBMS on a different machine that we wish to name County parcel records on main server or some other useful name that will help us immediately recognize that data source.

In this manner we can add more data sources. When done adding data sources, we can highlight the one desired and press OK.
Back in the Database Console we can see we have connected to the local SQL Server Express database. This database has very little data in it, only three drawings. Note that Database Console will display the friendly name we assigned to this data source.

**Example: Create a Data Source for an Access .MDB File**

Let's add a connection to an Access .mdb database file. We'll connect to the sample Northwind Traders file distributed with many editions of Access.

We begin by launching Database Console and then clicking the [...] button to specify a data source.

Click the Add Data Source button in the Data Source dialog.

In the **Connect To** dialog choose **MDB Files** in the **Files of type** box. Navigate to the folder and file desired and double-click on the file, in this case the nwind.mdb file.
This adds another OLE DB data source, since Manifold automatically connects to files such as .mdb using Microsoft's built-in OLE DB provider for such files. To avoid confusion we should change the name of this data source to something more recognizable.

We change the Name to Northwind Traders Sample MDB and press OK.
The result is that Database Console connects to the Northwind Traders database in the nwind.mdb file.

**Example: Create a Data Source for Oracle Express**

Let's add a connection to a local Oracle Express data source to the Data Sources dialog.

- We begin by launching Database Console and then clicking the [...] button to specify a data source.

- Click the **Add Data Source** button in the Data Source dialog.

In the **Connect To** dialog choose **Oracle Data Sources ()** in the **Files of type** box. This will connect using OCI, the **Oracle Call Interface** native to Oracle.
In the Oracle Data Source dialog we provide the Server name, the User Name and the Password. Settings shown are the default settings recommended in Manifold’s installation instructions for Oracle Express. As always, pressing the Test button to make sure we have entered the password and other information correctly is a good idea. When we are happy the connection is live, we press OK.

Back in the Data Sources dialog we see that a new data source has been added using the default name for the connection and showing that it is a native Oracle connection. We will add a friendly name.
In the **Name** box we enter **Local Oracle Express**. This will help us remember that this particular Oracle data source is a connection to the local DBMS, which will be useful if in the future we create connections to Oracle databases on different machines. Press **OK**.

Back in the **Database Console** we can see we have connected to the local Oracle Express database. (We could use some work on the filter settings in Database Console for this particular connection to suppress display of all the various DBMS objects that are not Manifold components.)

**Linking Drawings or Surfaces**

The **Data Source** dialog is *always* used when linking drawings or surfaces. For example, if we would like to link a drawing from an **MDB** file (such as when creating a linked drawing from a geocoded table), we must first add the **MDB** file as a data source in the **Data Source** dialog and then link the drawing from that **Data Source**.
In addition, when linking a drawing or a surface the Data Source dialog will launch with a "built-in" data source of This Project. Choosing that data source will connect to the project and make available components within the project from which the drawing or surface can be linked. The primary usage will be linking a drawing or surface from a table or query within the project.

**Connection Technologies**

There are many different connection technologies in use in the Microsoft world that we can use to connect to data sources.

When connecting to a native spatial DBMS we must always use the native connection technology if one is available. For example, when connecting to Oracle databases we should always use OCI, the Oracle Call Interface.

Use OLE DB to connect to databases when using generic Manifold spatial indices for spatial DBMS work or when connecting to SQL Server 2005 using the Manifold Spatial Extender for SQL Server or when connecting in a general way to databases. The disadvantage of using OLE DB is that the dialogs required to initiate a connection are more complex. Given that the Data Source dialog remembers previous data sources added to the dialog's list, that is not inconvenient because once a connection is set up it can be used with a simple double-click in the Data Sources dialog.

**Connecting via ADO.NET**

Manifold can also import or link data from ADO.NET data sources. Manifold includes a generic importer that is listed as ADO.NET Data Sources() in the Files of Type box when creating a new data source in the Data Source dialog. In addition, Manifold includes four dedicated importers designed for importing or linking data from the standard ADO.NET providers installed by the .NET Framework: ODBC, OLE DB, Oracle and SQL Server.

The generic import prompts for a .NET DLL (in the Assembly box) with an ADO.NET provider, the provider class, and the connection string.

The ADO.NET ODBC importer displays the standard ODBC data source dialog. The ADO.NET OLE DB importer displays the standard OLE DB data source dialog. The ADO.NET Oracle importer prompts for a server name and login credentials (integrated security or a specific login and password). The ADO.NET SQL Server import prompts for a server name, login credentials (integrated security or a specific login and password), and optional database name.
Some UDL (Universal Data Link), DSN, OLE DB and ODBC data sources may be dependent on the presence of third party database systems or ODBC drivers in the Windows system. For example, to connect to Btrieve databases, we will need a Btrieve ODBC driver on our system. To connect to Oracle databases, we will need an Oracle database server to which we can connect. Manifold automatically installs a variety of OLE DB providers that will connect to most popular data sources.

**Important:** Tables linked from ADO.NET data sources are always read-only. If read-only access is acceptable, ADO.NET is often the fastest possible connection and should be used in preference to ODBC or OLE DB for that reason. If read-only access is not acceptable, then OLE DB should be used if possible.

### Oracle Call Interface

When connecting to Oracle data sources Manifold can use the Oracle Call Interface (OCI), the native Oracle interface for exchanging data. Using the native interface provides better performance and allows using features not exposed through generic database interfaces such as ODBC, OLE DB and ADO.NET.

Exchanging data with Oracle data sources via OCI automatically maps Manifold geometry columns into Oracle SDO_GEOMETRY data. SDO_GEOMETRY values representing collections of geometric entities are not supported.

### Data Link Properties Dialog

When configuring a data source, Manifold uses the standard Microsoft OLE DB data source Data Link Properties dialog to allow connection to OLE DB data sources. It's a lot easier to use than it looks.

Choose the OLE DB data link type desired. Note that in Microsoft's world there are often many different ways to access the same type of data. For example, Manifold can open any Access .mdb file directly, or we can open an .mdb file by using the Microsoft Jet 4.0 OLE DB Provider. If we've installed any "Office" software or other database capable software, we will also have an ODBC driver on our system that can open Access .mdb files as well. So, we could use the OLE DB Provider for ODBC Drivers to open an ODBC data source for .mdb files.

After choosing the provider, click the **Next** button to move to the **Connection** tab.
The **Connection** tab will be pre-loaded with the right connection parameter choices for the type of OLE DB provider selected. In the illustration above, we've chosen the provider for ODBC Drivers and so we see the right connection options for connecting via ODBC. The **data source name** list box will be pre-loaded with the ODBC data sources currently configured on our system.

If we had chosen the OLE DB Provider for SQL Server and then pressed **Next**, the **Connection** tab would be loaded with the right choices to connect to a SQL Server database via OLE DB.

Provide the necessary information for the provider chosen and then press **OK**. The **Advanced** tab shows various advanced options for the specified provider, and the **All** tab provides a summary of the data link properties specified.

Sophisticated database systems such as SQL Server or Oracle may be configured in sophisticated ways. The information in the **Connection** tab and, possibly, the **Advanced** tab reflect the sophistication of such systems. If you are unfamiliar with the specific expectations of the SQL Server or Oracle or other provider that you wish to use, you may have to consult with the database administrator in charge of the installation to determine the correct settings to use.

**Connecting Via ODBC**

ODBC as a database connection technology preceded OLE DB, so there are more ODBC drivers for different types of databases than there are OLE DB drivers. Manifold System connects to ODBC data sources through whatever ODBC drivers are installed on the computer.

ODBC is easy to use if we remember that the slightly tedious part is creating a new "source." Some ODBC sources are quite simple and consist of specifying what type of database driver to use (dBase, Access, etc.).
Other ODBC sources, such as connecting to SQL Server via ODBC, are more complex and require what type of driver to use, which database is to be opened and from which machine or server it is fetched. In the case of some systems, such as SQL Server and Oracle, there are both OLE DB and ODBC drivers.

Set up ODBC data sources using the Windows ODBC Data Sources administrator dialog that is called by clicking on the ODBC Data Sources icon in the Windows Control Panel. This icon is located in the Administrative Tools folder in the Control Panel in Windows 2000 and subsequent Windows editions.

Using UDL Files and OLE DB Data Sources

A Microsoft UDL file is simply a pre-packaged shortcut to an OLE DB data source. We create a UDL file as noted below. We can then use OLE DB data link properties dialogs to configure the data link that will be used by the UDL file. Note that when opening an OLE DB data source or creating a UDL file we end up using exactly the same OLE DB dialogs.

Why bother with creating a UDL file if doing so requires us to use the OLE DB dialogs anyway? It's not really necessary with the Data Sources dialog, which can remember an OLE DB connection perfectly well. However, creating a UDL file can serve as a shortcut for other applications to connect to the same data source.

Creating and Using a UDL File

Creating a UDL file is a standard Microsoft procedure that is documented in Windows (search for "UDL" in Windows Help). Suppose we have an Excel spreadsheet that contains a table and we would like to connect to that table using a UDL file. Our spreadsheet is called MySheet.xls. We would proceed as follows:

1. In Windows Explorer, right click and choose New - Text Document. Create a text document called MySheet.udl. Windows will complain about changing the extension from .txt to .udl. That's OK.
2. Double-click on MySheet.udl to open the Data Link Properties dialog.
3. Configure the Data Link Properties dialog as shown in the illustration below. Press OK. The UDL will now contain a data link to the Excel spreadsheet.
4. Launch Manifold and choose File - Import - Table and then open MySheet.udl. Choose the table desired.
UDL files are normally used to automate more complicated connections than shown above. They may be used to provide user credentials such as a user name and password or to use a complex connection string when required. Windows 2000, Server 2003 and XP all include documentation for creating UDL files as part of Windows help. Other versions of Windows may not include this documentation, although the capability is there if you have installed any one of many packages that install the Microsoft data access routines that enable universal data links.

Performance

The internal Manifold database engine used to maintain imported tables is usually substantially faster than external database systems, and is matched or exceeded only by the very fastest enterprise class DBMS products such as Oracle or SQL Server.

Manifold’s facility with Microsoft data types as well as installation remarks for some Windows versions admonishing users to install Jet service packs occasionally leads to the misconception that Manifold internally uses Jet for database management. That is not the case.

Although Manifold includes a copy of Microsoft’s “Jet” database engine (the same used in Access), Jet is not used to maintain tables that are imported into Manifold. Jet is used only as an accessory to handle external files. All imported tables are managed using the internal Manifold database engine and all queries are executed using the Manifold SQL engine, both of which are purpose-built systems engineered for the specific demands of fast geometry/attribute work in GIS.

When linking external tables into Manifold one ends up using whatever database engine or system is associated with those external tables. Performance will therefore almost always decrease when using linked tables from simple, consumer-style data sources (such as Access or Excel tables) as compared to the speed of imported tables.

Linked tables will be as fast as or faster than internal Manifold tables when the external tables are very large and the provider is a sophisticated engine such as SQL Server or Oracle using a fast connection such as ADO.NET (or, in some circumstances, OLE DB) for SQL Server or OCI for Oracle. Even if they are not as fast as Manifold’s
dedicated engine for smaller databases, enterprise database systems such as SQL Server nonetheless are still very fast. Therefore, it is not likely that any performance difference between imported tables or linked tables will be noticed with smaller tables when enterprise class engines such as Oracle or SQL Server are used.

When using external DBMS tables, use the fastest connection possible, which may vary depending on the DBMS in use. For example, when connecting to Microsoft's SQL Server, ADO.NET is faster than OLE DB and OLE DB is faster than ODBC. ADO.NET is much faster than ODBC. However, ADO .NET is generally read-only, which limits the utility of that technology. Consider that connecting to a remote SQL Server can take much longer than connecting to a local SQL Server. For example, suppose that connecting to a table inside the Manifold project takes 1 unit of time. A very rough guide to equivalent times to access the same table using different connection and DBMS technologies is listed in the accompanying table. Clearly, it is very unwise to connect to remote SQL Server installations using ODBC.

<table>
<thead>
<tr>
<th>Time</th>
<th>Source for Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Native table (inside the project)</td>
</tr>
<tr>
<td>8</td>
<td>Linked from a local MDB file</td>
</tr>
<tr>
<td>10</td>
<td>Linked from a local SQL Server via ADO.NET</td>
</tr>
<tr>
<td>40</td>
<td>Linked from a local SQL Server via OLE DB</td>
</tr>
<tr>
<td>10</td>
<td>Linked from a remote SQL Server via ADO.NET</td>
</tr>
<tr>
<td>250</td>
<td>Linked from a remote SQL Server via OLE DB</td>
</tr>
<tr>
<td>6000</td>
<td>Linked from a remote SQL Server via ODBC</td>
</tr>
</tbody>
</table>

The default connection to Oracle servers is always OCI, so Oracle connections always run at the fastest possible connection speed unless we manually force the connection to use an alternative, such as OLE DB or ODBC.

See the Performance Tips topic for specific tips to maximize table and query performance.

**Working with SQL Server and Oracle**

**Very important:** When working with server-based OLE DB providers such as SQL Server and Oracle, users are strongly encouraged to maintain primary keys in all tables linked into the Manifold project. A side effect of how such servers interact through OLE DB is that if the table does not have a primary key, performance will be greatly reduced.

**Linking to Oracle Tables**

The default connection to Oracle tables is OCI, which greatly simplifies life with Oracle. Connect using the Oracle Data Sources type and all will be well.

However, if for some reason we need to connect using OLE DB or ODBC to an Oracle Server, we should make sure to follow these tips:

- Always set the **Allow saving password** option. If this option is not set, the system will connect for the first time but not at all other times. The intricacies of the OLE DB / ODBC interaction that takes place within the Microsoft OLE DB Provider for ODBC Drivers are such that it is almost never possible for the Manifold to jump into the middle of the connection process and prompt the user for a username and password.

- Do not use the native Oracle ODBC driver. Use the Microsoft OLE DB Provider for ODBC and use Microsoft ODBC for Oracle. The native Oracle drivers are limited to forward-only cursors (which is by far the simplest and the lowest performance type of database cursor allowed in ODBC drivers) and thus are unusable in Manifold or many other programs, such as Microsoft Access. Future editions of Manifold will add special routines to allow use of the Oracle driver.

**Working with OLAP**
OLE DB drivers allow connections to many data sources besides simple database tables. For work with OLAP, for example, Microsoft's SQL Server Decision Support Services (the Microsoft OLAP package) has an OLE DB that can be used to import or link a "table" that pulls data from the OLAP cube. The driver supports Microsoft's MDX language that can be used to fetch information from OLAP cubes. The MDX language is very rich and thus allows sophisticated work with OLAP cubes. Needless to say, the driver works fine with Manifold.

When Logins and Passwords are Required

When linking tables via a connection to some database providers such as Oracle or SQL Server the connection may fail without proper user credentials if security settings in the database provider require a username and password. In such cases Manifold will raise a login dialog that allows specification of a user name and password.

The dialog includes a Remember username and password checkbox that is off by default. Check this box to save the provided user name and password with the table within the .map file. This will enable automatic linking using these credentials the next time the .map file is opened. Note: the user name and password will be saved with the table in encrypted form within the .map file. It is strongly recommended that any such .map files containing your user credentials be protected by setting appropriate Windows access privileges on the .map file or the folder that contains it.

Tech Tips

Manifold can use all of the standard Microsoft DBMS connection technologies, such as ADO .NET, to connect to data sources. There are a huge number of possibilities with such connection technologies that are documented by Microsoft and other educational resources for standard Microsoft facilities. As this documentation is aimed at Manifold and is not intended as a general educational resource for standard Microsoft facilities it will not attempt to duplicate the many educational resources already available for those standard Microsoft technologies.

If you would like to utilize databases such as SQL Server or Oracle and you would like to connect to them using technologies such as OLE DB or ADO.NET, there are very many books on those subjects that provide full details, tips and other useful information on using these standard connection technologies.

The data source selection dialog automatically recognizes data sources which do not support storing drawings, images or surfaces. For example, we cannot store images into a .txt file. Attempting to select a data source which does not support storing components of a certain type during the export of a component of that type displays a warning message below the list of data sources and disables the OK button.

Notes

In a perfect world, every OLE DB provider or ODBC driver or external DBMS they serve that is installed on our systems would work without error. Regrettably, in real life the various database drivers and providers installed in our systems will contain bugs. Even in the case of the high quality drivers provided by Microsoft, one only need read the Microsoft Knowledge Base and the release notes for various Microsoft service packs to see that many bugs have been identified in such drivers. Manifold uses Microsoft drivers and OLE DB providers to connect to various types of database tables. If there are errors in the Microsoft drivers, such errors will affect Manifold as they would any other application.

If you suspect a database driver bug, take time to research the issue within the online Microsoft Knowledge Base. A useful experiment might be to import the table instead of linking to it and seeing if the problem persists. If it goes away, that is evidence that the problem lies in the external database system or is a bug in the drivers for that system. To reduce the likelihood of bugs in external software make sure you have applied the latest service pack for the external database system you are using.

When running SQL within Manifold queries, one is using the Manifold SQL engine. When executing SQL within the Database Console one is using whatever SQL is the native SQL of the external database system. One should be aware that SQL implementations in various database systems can contain numerous bugs. For example, even as well crafted an SQL as Jet SQL used within Microsoft's Access products contains numerous bugs. If an SQL bug occurs within the Database Console, the bug should be tracked down with the vendor of the external database system being used.

See Also

Tools - Administrator Console
Tools - Database Console
NVIDIA CUDA

For many operations Manifold will automatically use multiple processors or multiple processor cores if installed in a computer system. In addition to this basic multiprocessing capability Manifold includes the ability to utilize massively parallel multiprocessing in the form of NVIDIA CUDA-enabled products, such as the NVIDIA GPU plug-in card seen below that provides 480 processors and supercomputer performance for under $500.

It is not an exaggeration to say that NVIDIA CUDA technology could well be the most revolutionary thing to happen in computing since the invention of the microprocessor. It's that fast, that inexpensive and has that much potential. NVIDIA CUDA is so important that all Manifold users should insist that computer hardware they procure is CUDA-enabled. In fact, given the ubiquity of NVIDIA products it is quite likely that a recently-procured computer will already include a CUDA-enabled NVIDIA GPU of some sort.

Introduction

NVIDIA is best known for motherboard chip sets as well as for outstanding graphics processors that have become popular as the basis for graphics cards. In the quest for maximum speed, NVIDIA’s GPUs (Graphics Processing Units) have evolved far beyond single processors. Modern NVIDIA GPUs are not single processors but rather are parallel supercomputers on a chip that consist of very many, very fast processors. Contemporary NVIDIA GPUs range from 16 to 480 stream processors per card, delivering incredibly powerful computing bandwidth. The card shown above, for example, provides 480 stream processors.

Although the market impetus behind the creation of such supercomputers on a plug-in board has been the computational demands of the PC gaming market, such "graphics" boards have become so powerful that the scientific computing community has begun using them for general purpose computing. It turns out that many mathematical computations, such as matrix multiplication and transposition, which are required for complex visual and physics simulations in games are also exactly the same computations that must be performed in a wide variety of scientific computing applications, including GIS.

NVIDIA has supported this trend by releasing the CUDA (Compute Unified Device Architecture) interface library to allow applications developers to write code that can be uploaded into an NVIDIA-based card for execution by NVIDIA’s massively parallel GPUs. This allows applications developers to plug in a teraflop-class, 480-processor, NVIDIA-based card and upload applications to run within the NVIDIA GPU at far greater speed than possible on even the fastest general purpose CPU on the motherboard. For a mere few hundred dollars we can use CUDA to achieve true, supercomputer performance on the desktop.

CUDA offers such tremendous performance gains that many functions within Manifold have been re-engineered to execute as parallel processes within CUDA if such a card is available. If we have a CUDA-capable NVIDIA graphics card installed in our system, Manifold can take advantage of the phenomenal power of massively parallel NVIDIA stream processors to execute many tasks at much greater speed.

Because NVIDIA technology benefits from enormous economies of scale in the gaming market, CUDA-enabled cards have become very inexpensive for the performance they provide. At the present writing CUDA-enabled cards can be purchased for less than $100 for an entry-level CUDA-capable card and easily under $350 for a high performance CUDA-capable card. It is easy and inexpensive to choose a card with the balance between performance and cost desired (more stream processors running at faster clock rate with more memory gives better performance).
CUDA-Enabled GPUs

Dozens of vendors provide graphics cards based upon CUDA-capable NVIDIA GPUs and it is almost not possible for a high-performance PC or motherboard vendor to introduce a product that does not do a good job of hosting such GPU cards. The insatiable demand of gamers for more performance has also spawned an industry of vendors offering ever-faster memory, more powerful power supplies and other systems components that are perfect for creating outstanding GIS desktop and server machines.

The easiest way to see if a particular graphics card is CUDA-enabled is to first check which NVIDIA GPU it utilizes. Next, visit the NVIDIA web site at http://www.nvidia.com and in the web site's search box enter "CUDA-Enabled GPU" to find pages that list NVIDIA GPU products which can be utilized for CUDA-enabled parallel processing. Almost all contemporary NVIDIA GPUs are CUDA-enabled.

If anything, the surprise is discovering that quite a few NVIDIA GPUs aimed at motherboard chipsets or mobile applications like portable computers are also CUDA-enabled, albeit with a smaller number of processors per GPU.

NVIDIA at the present writing provides three families of CUDA-enabled GPU products. All three families may be used with Manifold and CUDA:

- **GeForce** - The GeForce line of NVIDIA GPUs are sold primarily through a wide variety of graphics card and motherboard manufacturers which incorporate the NVIDIA chips into their own graphics cards. Performance tends to be high and prices kept low by fierce competition in gaming markets.

- **Quadro** - The Quadro line of NVIDIA GPUs is manufactured and sold directly by NVIDIA into very high end professional workstation graphics markets. Quadro cards provide extraordinarily high resolutions and massive graphics memory for the most demanding workstation applications. Some Quadro products also appear in high end portable computers, and some Quadro products are also provided in external cabinets similar to Tesla packaging.

- **Tesla** - The Tesla line of NVIDIA GPUs is also manufactured and sold directly by NVIDIA to support high performance computing where supercomputer performance through parallel processing is required. Although they are also available as plug-in cards, Tesla GPUs are best known for being packaged into external cabinets (either desktop or rack mount) that provide two or four GPUs per cabinet.

The external cabinets used for Tesla and some Quadro products attach to desktop computers using a special cable that plugs into an interface card plugged into a standard PCI-E slot. This allows the external Tesla or Quadro configuration to appear to software as if it were a plugged-in PCI-E card just like typical GeForce cards. However, because the actual GPUs are hosted in an external cabinet that provides power and cooling, the host computer does not need to be retrofitted with additional power and cooling.

CUDA Limitations and Requirements

There are several important constraints on CUDA use within Manifold:

- We must have a CUDA-enabled NVIDIA card installed in our system. 200 and 400 series NVIDIA cards at the present writing are the best-known CUDA-enabled cards, but other NVIDIA GPUs are also CUDA-capable (check with the NVIDIA web site and with your graphics card vendor's web site to see if a particular card is CUDA-capable). Hardware evolves so rapidly under the pressure of gaming industry economy-of-scale that almost before this documentation can be published there will be even faster CUDA-capable cards. manifold.net recommends getting Fermi-class (400 series) GPUs. May as well get the best!

- The rest of our PC system must have sufficient speed and power to support the NVIDIA card. For example, memory must be fast enough to handle CUDA bandwidth and power supplies must provide enough power to run the NVIDIA card (or cards) with extra PCI-E power cables. Consult any technology-obsessed, 14 year old gamer for advice on configuring a suitably "hot" system.

- We must have installed NVIDIA's most recent set of drivers for Windows, which may be downloaded from the nvidia.com web site. NVIDIA's latest drivers automatically install software required for CUDA use by CUDA-capable NVIDIA-based cards.

- If we are running a 64-bit Windows system we must have installed NVIDIA's 64-bit, CUDA-enabled drivers for our 64-bit Windows system.

- Writing massively parallel algorithms to implement spatial functions is extremely difficult, even for manifold.net. Therefore, at the present time only a few dozen functions have been implemented within Manifold that can leverage CUDA. Many more are on the way.
Introduction

- Existing CUDA-enabled functions within Manifold are Surface - Transform dialog operators for surfaces. The Surface - Transform dialog is part of the optional Surface Tools extension for Manifold (and also a built-in part of some Manifold System editions such as Universal Edition and Ultimate Edition). If we do not have the Surface Tools extension we will not have the ability to use this dialog and hence no ability to leverage CUDA. New updates and future Manifold releases will likely add many more usages of CUDA in addition to the Surface - Transform dialog operators.

- Functions executed within CUDA cards are virtually instantaneous compared to speed of execution within the main processor. However, the NVIDIA stream processors execute tasks so rapidly that it is difficult to provide data fast enough from disk and memory to keep the processors occupied. The resulting performance in most “real life” applications therefore tends to be limited not by processor speed but rather by the speed with which data can be fetched from hard disk or other memory. In addition, a good portion of various tasks are not bound by computation but instead involve overhead tasks such as writing out results to disk, re-computing levels and other necessary but mundane tasks that are not accelerated by CUDA processors. The net result is that as a practical matter for many tasks CUDA-enabled processors will visibly increase speeds, almost always by a factor of two to ten and at times by a factor of ten to fifty, but not usually by factors of hundreds for the overall task even if the actual computation of parts of the task goes hundreds of times faster.

- We can get the most out of CUDA if the rest of our machine does not slow down the ability to feed the insatiable power of NVIDIA stream processors. For maximum speed we should use 64-bit Windows on at least a quad core machine with lots of RAM and large, fast disk drives. Before configuring a new 64-bit system, check the NVIDIA web site to make sure that 64-bit drivers are available for the Windows operating system you plan to install. At the present writing, Windows XP x64 has been used as a baseline common denominator for development of x64 support by manifold.net with all development in Vista and Windows 7 as well.

Despite the above limitations it is clear that CUDA is a revolutionary technology. NVIDIA GPUs are so fast that a routine comment from developers is that NVIDIA renders the main processor almost superfluous, as if even the fastest multi-core Intel chip is relegated to being nothing but an accessory processor to handle the keyboard and mouse. That is not hyperbole given that NVIDIA GPUs can run jobs 200 or even 300 times faster than even the fastest Intel CPUs. See the demo below for an example.

Such speed advantages are not a competitive challenge that traditional processor vendors can afford to ignore. CUDA is the first of what is likely to be a new wave of massively parallel architectures from competitors such as Intel and AMD. Manifold's parallel code has been expressly written to allow easy implementation on future “many-core” processor solutions from Intel and AMD that will compete with NVIDIA CUDA.

Installation and Configuration

Once we have installed CUDA-capable NVIDIA hardware and NVIDIA drivers there is no need for any other configuration. Note that you must install 64-bit NVIDIA drivers when operating 64-bit Windows.

Make sure to download and install the latest NVIDIA drivers. Current generation NVIDIA software installs CUDA capability as part of the main NVIDIA driver installation. Manifold looks for current NVIDIA software. Because driver installation discs provided within graphics board packages might have been mastered many months ago, it is important to download and install the latest drivers from NVIDIA's web site.

Make sure to download and install the latest Manifold update. Although the nature of CUDA means that few changes need be made to support new NVIDIA GPUs, some changes are required to keep up with rapid evolution. Recent updates have added support for Fermi-series NVIDIA cards, for example. If you don't install the latest Manifold update you won't be able to use Fermi-series cards.

When launched, Manifold will automatically detect and utilize CUDA-enabled hardware. The Use GPGPU technologies (NVIDIA CUDA) option in the Tools - Options - Miscellaneous dialog is turned on by default.

When a CUDA-enabled card is present Manifold will report finding the card in the Help - About dialog in the GPU value. Manifold System Release 8.00 reports CUDA-enabled GPUs either as the more recent Fermi series GPUs or all earlier CUDA-enabled GPUs as pre-Fermi.
The above illustration shows a Manifold Help - About report in a 64-bit Windows system in which two CUDA-enabled cards have been installed, both of which use NVIDIA GPU devices prior to the Fermi series.

If a CUDA-enabled card has not been installed or if current NVIDIA drivers have not been installed or if the Use GPGPU technologies (NVIDIA CUDA) option in the Tools - Options - Miscellaneous dialog has been turned off, the Help - About dialog will report Graphics only for the GPU, as seen above.

If a Fermi series card is present, it will be reported as seen above. All Fermi series cards (for example, GTX 480 or GTX 470 GeForce cards or Tesla C2050 or C2070 cards) will be reported as a Fermi device. The above shows a Help - About display for a 64-bit Windows system that has one Fermi card, a GTX 480, in it.

Functions Utilizing CUDA

At the present writing the following Manifold Surface - Transform dialog functions utilize CUDA if available: Aspect, AvgValue, Blur, CurvGauss, CurvMean, CurvPlan, CurvProfile, DifferenceE, DifferenceN, DifferenceNE, DifferenceNW, DifferenceS, DifferenceSE, DifferenceSW, DifferenceW, Diversity, DiversityIndex, HighPass1, HighPass2, HighPass3, Laplace1, Laplace2, LowPass1, LowPass2, LowPass3, MajValue, MaxValue, MedianCross,MedianSquare, MedianSquare5, MedValue, MinValue, Sharpen, SharpenMore, Slope, SumValue, Tile and TileMedian functions.

Additional functions and use of CUDA are expected to be added with each new Manifold release. New products such as the successor to Release 8 expected in 2010 will include new internal architectures to support CUDA even more effectively.

Demos

We often will be asked to demonstrate the speed of CUDA for colleagues, so it is handy to have a familiar example available that can be quickly run to demonstrate the power of GPU computing. This demo uses the example Montara Mountain surface within an easily-remembered demonstration that shows the power of CUDA.

This example assumes we are running 64-bit Windows XP on a machine with at least one CUDA-capable NVIDIA graphics cards installed. It will work fine on 32-bit Windows as well, but why use old-fashioned Windows when 64-bit Windows has been happening for over seven years? It also will work fine on Windows 7 or Vista or other Windows versions. In this particular example we've installed one GTX 480 card, which uses a Fermi series GPU that provides over 480 processing cores.

The demo also requires a Manifold installation that includes the Surface Tools extension (automatically enabled when Universal Edition or Ultimate Edition are installed) so that the Surface - Transform dialog is available.

The greatest performance difference visible with CUDA appears when the Surface - Transform dialog is used for a complex calculation on a surface that is not too large. This shows off the intense computational performance delivered by CUDA without requiring many disk accesses (which take a proportionately larger amount of time for large files) slowing down the works.

For an example of a complex calculation, suppose we have a surface called MySurface. Launch the Surface - Transform dialog and execute a formula such as:

\[ \text{Slope([MySurface]) + Slope([MySurface] * 2) / 2} \]
Slope([MySurface] * 3)/3 + Slope([MySurface] * 4)/4

This is a nonsensical but complex formula that will execute much faster using CUDA than without CUDA. Although this and the other formulas shown in this example are artificial examples, they share many of the mathematical characteristics of sophisticated "real life" operations on surfaces so they are genuinely representative of real performance gains available through CUDA.

The more complex the formula the greater the advantage from using CUDA. For example, we can compute aspect as well as slope and also use an optional window size parameter for both functions in a formula (using a window size of 5) such as:

\[
\]

Let’s apply the above to a specific example.

We will use the Montara Mountain sample surface to measure computing speed with and without CUDA. We begin by importing the Montara Mountain surface, as is illustrated in examples such as the Combine a Surface and a Drawing in a Map topic.

We rename the surface to a very short, one letter name, s, to facilitate quick keyboarding.
In the Tools - Options dialog's Miscellaneous page we verify the Use GPGPU technologies (NVIDIA CUDA) option has been turned on.

Manifold can log reports to the history pane such as the time required to execute functions. We will also turn on the Log transform time option in the Logging page in the Tools - Options dialog.

Launching the Help - About dialog we can see that Manifold has detected a Fermi series CUDA-capable device installed in the system.

CPU: Intel(R) Core(TM)2 Quad CPU @ 2.40GHz
GPU: CUDA (1 device, Fermi)
RAM: 6190 MBytes
OS: Windows XP (b3790) Service Pack 2
Running in 64-bit mode.

No doubt even a fire-breathing GTX 480 as used in this example with 480 processors per device will soon seem slow given how fast NVIDIA is introducing newer and more powerful devices, but this example using a GTX 480 is still highly instructive for the tremendous gains achievable through GPU computing at very low cost.
With the surface open in a window we launch the Surface - Transform dialog entering the following formula (which may be copied from this Help window and pasted into the dialog):

\[
slope(s,5) + aspect(s,5) + \nonumber \\
slope(s*2,5) + aspect(s*2,5) + \nonumber \\
slope(s*3,5) + aspect(s*3,5)\nonumber
\]

We check the Save result as new component box for two reasons: first, because that uses slightly less overhead than applying the results to the subject surface and, second, because not altering the subject surface preserves it unmodified for the next trial.

Press OK and Manifold launches into action using CUDA to compute the formula, reporting the time required in the History pane:

Surface Transform: 1.494 sec

The History pane reports how long the computation takes, in this case approximately 1.5 seconds. Times may vary slightly depending on precise system configuration: a Fermi series GPU providing 480 stream processors is so incredibly fast that it computes the task almost instantaneously. The time required overall is mostly overhead such as fetching data from disk and getting it through the CPU to the GPU and not actual calculations; therefore, running the same task again will cut the time slightly by about a tenth or a few hundredths of a second as Windows caches the data used for more efficient provision to the GPU.

The result of the Surface - Transform formula computation is a new surface, automatically named \( S_2 \).

Let's now see how long the computation takes without using CUDA. We can measure this in identically the same system for an “apples to apples” comparison by simply switching off the Use GPGPU technologies (NVIDIA CUDA) option in the Tools - Options dialog.

In the Tools - Options dialog's Miscellaneous page we uncheck the Use GPGPU technologies (NVIDIA CUDA) option to instruct Manifold not to use CUDA.
Launching the Help - About dialog again we can now see that Manifold no longer uses CUDA and reports
Graphics only for the installed GPU.

Once more we run the same formula in the Surface - Transform dialog. As before, the time required for the
computation is reported in the History pane, appearing below the last time logged:

Without CUDA the computation takes dramatically longer, over 424 seconds. Wow! Using the CPU instead of the
NVIDIA Fermi GPU took about 284 times longer. It takes so much longer because the computation must happen
on the Intel main CPU, which even when run as a true 64-bit device by 64-bit applications code in 64-bit Windows
is far, far slower than the phenomenal supercomputer speed of the NVIDIA Fermi device. In this case, the main
CPU really is insignificant compared to the speed of the GPU.

It's true the machine used in this example is not the latest, greatest machine: it has an Intel Core 2 Quad CPU
and not the latest Core i7 quad. An Intel Core i7 is indeed a faster CPU than the prior generation Intel Core 2
Quad used in this example. However, most users do not yet have a Core i7, and even if they did the Core i7 is
not remotely as fast as the NVIDIA GPU. If we spend a lot more money to buy a really fast Core i7 we can drop
the time required from about 424 seconds to about 261 seconds, that is, only about 1.6 times faster than the Core
2 Quad instead of being 284 times faster with the NVIDIA GPU. That's no comparison, especially when faster
Core i7 processors cost more than the $300 to $500 cost of a Fermi card. Even a less expensive Fermi card like
a GTX 470, selling for under $350 as of this writing, will execute the job in only 1.7 seconds, about 250 times
faster than the CPU alone.

More complex formulas will require yet longer times. For example, in the same system used above, computing
the formula...

\[
\text{slope}(s, 5) + \text{aspect}(s, 5) + \\
\text{slope}(s^2, 5) + \text{aspect}(s^2, 5) + \\
\text{slope}(s^3, 5) + \text{aspect}(s^3, 5) + \\
\text{slope}(s^4, 5) + \text{aspect}(s^4, 5)
\]
...requires \(2.389\) seconds with CUDA and \(740.955\) seconds without CUDA, about \(310\) times faster with CUDA.

The comparison is especially dramatic when considering that the hardware used for the above example is a typical, highly capable base system with a 64-bit, quad core processor and 8 GB of RAM. It's faster than the dual core CPUs used by most people. A fast system makes for a good demo, because it minimizes the time required for overhead chores accomplished by Manifold as part of the demo. The actual computation of surface values using CUDA is very fast with most of the time required for the CUDA-enabled timings going to overhead such as setting up the job and writing out the resulting surface.

**Very Important:** After doing this demo, don't forget to turn on the Use GPGPU technologies (NVIDIA CUDA) option in the Tools - Options dialog so that future work can take advantage of CUDA!

**Demo Notes**

Experienced demonstrators will usually create the above demo in advance as a Manifold .map project file that has the surface already imported and example computations for the Surface - Transform dialog saved in Comments components as text (a .map file with all that done may be downloaded from the manifold.net web site).

To do the demo, the Comments component can be quickly opened and the desired text copied and then pasted into the Surface - Transform dialog. Showmanship is an important part of good demos so it is important not to allow the audience to get bored while we keyboard a formula into a dialog. Using copy and paste also eliminates the need to remember the exact syntax of slope or aspect functions. The Surface - Transform dialog will "remember" the last formula used in a .map project, but just in case someone changes the formula it is a good idea for demos to have a spare copy of the formula in a Comments component.

Part of showmanship is launching the Help - About dialog after changing the option to use or not use CUDA so that the audience can see for themselves that Manifold is or is not using CUDA. This is the computer demo equivalent of a magician showing the audience "there is nothing up my sleeve."

It is also important to remember to turn on the Log transform time option in the Logging page in the Tools - Options dialog and to have the History pane open, so that the audience can see for themselves the exact timing of each trial.

Although longer demos can show the very much longer periods of time required for non-CUDA performance, it is important not to bore the audience. A comparison of \(1.5\) seconds to seven minutes usually conveys the intended message. \(1.5\) seconds goes by instantly, especially if the demonstrator says a few words about what is going on after pressing the OK button, while seven minutes seems endless and unendurable in comparison.

[The unendurable seven minutes make for an especially memorable demo for those who want to contrast Manifold's CUDA-enabled supercomputer speed with legacy GIS software products that do not have CUDA-enabled supercomputer speed but which do cost several thousand dollars per license to run hundreds of times slower.]

The easiest approach is to do the CUDA-enabled trial first, so the audience sees it goes rapidly, and then launching the interminably long process without CUDA on the second trial as the demonstrator discusses CUDA architecture, the wide availability of NVIDIA-based hardware, the breadth and depth of CUDA-capable devices and the extraordinary economics of scale attained by leveraging mass market interest in massively parallel GPUs for gaming. While all this talk is going on the audience will see with their own eyes that in the background the non-CUDA computation is still painfully crawling along for many minutes to do what the CUDA computation accomplished in a second or two.

More experienced demonstrators will often choose to perform the non-CUDA trial first. This requires greater skill with timing of presentations so that the speaker's commentary coincides more or less with the end of the computation. In this case the demonstrator will launch the transform operation and then spend the next six minutes or so talking about CUDA, describing what the Surface - Transform dialog is about, talking about the Montara surface, describing how even with a quad core main processor this is a long and complex calculation and
every now and then directing the audience's attention to the lengthy computation with a comment such as, "Nope, not done yet...". After such a set-up audiences are inevitably dazzled by the amazing speed of CUDA.

This particular demo uses a completely artificial formula that serves no practical purpose except to illustrate speed of computation. But the formula is very representative in that it provides an easily-understood example of a legitimately complex calculation that has about the same demands on computation of sophisticated mathematics typical of significant computations on surfaces.

"Real life" computations on surfaces tend to use very complex formulas that are difficult to explain to an audience that does not consist of remote sensing or GIS experts. However, just about everyone can understand what slope and aspect are.

**Slope** gives the degree of inclination of a surface. It is a way of finding which parts of a surface are relatively flat and which are steep. **Aspect** gives the directional orientation of a surface, to find which parts of a surface, for example, are facing South. Computing either slope or aspect involves a large number of local calculations over the entire extent of a surface to find the slope or aspect for each individual pixel of the surface. The use of a five pixel window simply controls how many neighboring pixels are considered when computing slope or aspect. The use of a window is a way to demonstrate additional computational complexity by forcing an interpolation of sorts within each local computation of either slope or aspect.

**Other Notes**

The **Help - About** dialog seen in this example shows that this particular system has one CUDA-capable device of **Fermi** class installed. The demo was run using an EVGA GTX 480 card providing a total of 480 stream processors. An outstanding benefit of NVIDIA technology is that speeds are doubling almost every year while costs are remaining constant or are going down.

The **Help - About** dialog reports the number of CUDA-enabled GPUs found. Some cards have more than one GPU in the "card." For example, some double-wide cards (such as the original GTX 295 reference design) are really two circuit cards packaged within the same double-wide fan housing, with a GPU on each card for a total of two GPUs. Such cards will be reported as two GPU devices. Plug in two such cards and Manifold will report four GPU devices. Some cards (such as the GTX 295 "Co-op" series) have a single circuit card with two GPU devices mounted on the card. Installing one such card will likewise be reported as two GPU devices.

Experienced Manifold users will see from the screen shot of the Montara Mountain surface that it has been projected and that a palette coloring the surface by height has been applied using the surface display options dialog. Projections and surface display options have no effect on performance of **Surface - Transform** computations with or without CUDA.

CUDA capability is enabled by default. To turn it off, turn off the Use GPGPU technologies (NVIDIA CUDA) option in the Tools - Options - Miscellaneous dialog. Remember to turn the option back on after doing a demo!

**Troubleshooting**

**No CUDA card reported in Help - About dialog**

- Has a CUDA-capable NVIDIA card been installed in the computer? Note that not all NVIDIA-based graphics cards have CUDA capability. You must have an NVIDIA-based card that supports CUDA. Almost all reasonably contemporary NVIDIA GPUs now support CUDA.
- Have you installed the most recent NVIDIA drivers downloaded directly from NVIDIA's web site? Older drivers, such as those often found on installation DVDs packaged with graphics cards, may not work even if they appear to install some sort of CUDA software.
- Have you changed a graphics card recently? Swapping cards, even within the same series (like switching from a GTX 470 to a GTX 480) may require re-installation of NVIDIA display drivers and restarting the system.
- If you are working with 64-bit Windows have you installed a 64-bit Manifold license? Have you launched the 64-bit Manifold installation? Recall that 64-bit Manifold installations will install both a 64-bit and a 32-bit Manifold executable so that a 32-bit Manifold version can be launched for compatibility with older, 32-bit software (such as Access); however, if you want to work with 64-bit CUDA in 64-bit Windows you should launch the 64-bit Manifold version. See the 32-bit and 64-bit Manifold Editions topic.
- If you are working with 64-bit Windows have you installed 64-bit NVIDIA CUDA drivers? Did you install the right NVIDIA drivers for your Windows system? For example, if you are running Vista x64 you should probably not expect that a driver package provided specifically for Windows XP x64 will work.
Note that as of this writing not all NVIDIA driver packages for all possible Windows versions include CUDA support. Drill down into the nvidia.com web site using search terms such as "CUDA downloads" to see if your Windows version is supported.

- Check to make sure that the Use GPGPU technologies (NVIDIA CUDA) option has been enabled in the Tools - Options - Miscellaneous dialog.

No Surface - Transform dialog

- Have you licensed the optional Surface Tools extension for Manifold and activated it? Launch the Help - About dialog to see if the extension is reported as an installed extension. The Surface Tools extension is a built-in part of Universal Edition and Ultimate Edition and does not require activation if you are using either of those two editions. If you are using some other edition and have licensed Surface Tools, you must activate it by following the instructions in the Installing and Activating a Manifold Extension topic.

- Do you currently have a surface open as the active window or as the active layer in a map? The Surface - Transform dialog is not available if the focus is not on a surface window or a surface layer in a map.

Performance gain not observed

- Launch the Help - About dialog: does Manifold report any CUDA-capable devices installed in the system? If not, see the troubleshooting section above.

- Are you executing functions within the Surface - Transform dialog? CUDA at the present writing works only with functions within that dialog. It makes no difference in other functions or other parts of the system, except of course that since CUDA cards are also very fast graphics cards the performance of 3D terrain rendering will be very good.

- Are you executing functions that are listed as supporting CUDA? Writing formulas in the Surface - Transform dialog that utilize functions not listed as supporting CUDA will not benefit from CUDA.

- Is the data set very small? Some cases of small data compute so rapidly that there is little to gain by using CUDA since even a base system without CUDA will execute the task very rapidly.

- Is the formula very simple? CUDA speeds up computation but not overhead like fetching data from disk. If a computation involves a simple formula there is not much to speed up because the computation will get done very rapidly in any event so the time for the job is mostly overhead.

- Is the data set very large? As with the case of simple formulas, if data sets are very large then overhead tasks like fetching data from disk become proportionately larger compared to the time spent on computation. Since CUDA increases the speed of computation, if computation is a proportionately smaller part of the task compared to overhead there will be less visible effect from CUDA.

- Does your project involve slow system resources? Read the Performance Tips topic carefully to make sure your project and your system are structured for maximum performance. If a project does something that cripples performance, such as using a linked component that must come into the project from a very slow network link, then overhead delays will be very great with or without CUDA.

- Are you sure CUDA has not improved performance? Try timing the task with CUDA on and with CUDA off by turning on and off the option to use CUDA in the Tools - Options - Miscellaneous dialog, as illustrated in the demo above. You may be surprised to find that something which seems to take forever, ten minutes or so, with CUDA indeed takes far longer, like hours or days, without CUDA.

See Also

Help - About
Performance Tips
Surface Tools
Surface - Transform
Tools - Options

Database Installations

Database Installations

Manifold System is often used with database management systems. In particular, Enterprise Edition is often used with DBMS packages to create Enterprise servers, or to store drawings in databases to allow concurrent, multi-user editing. Manifold traditionally has supported the three top commercial enterprise class DBMS packages, IBM DB2, Microsoft SQL Server and Oracle.
All three of these DBMS packages may be used as a spatial DBMS either using built in facilities (Oracle) or using optional, free spatial extenders (IBM and SQL Server). In addition, all three packages at the present writing are available in free "Express" editions that as of the time of this writing anyone may download from the vendor's website and use at no charge. See the vendor's web page for current availability and for details on the terms and conditions of use.

Because of free availability and superb functionality, these three packages (as well as open source alternatives such as PostgreSQL / PostGIS) are great choices for use with Manifold. The Express versions are for the most part equivalent to the flagship enterprise versions of each DBMS, with some reasonable limitations as follows:

**IBM DB2 Express-C Edition**
- Runs on up to two processors multithreaded using up to 4 GB of RAM. No limit on database size or on the number of users. Can be enabled as a spatial DBMS by downloading the free IBM Spatial Extender for DB2 from IBM's web site.

**Oracle Express Edition**
- Runs on one processor using up to 1 GB of RAM. 4 GB limit on database size. No limit on the number of users. Supports full Oracle spatial SDO_GEOMETRY technology. Does not support Oracle Spatial GeoRaster technology.

**SQL Server Express Edition**
- Runs on one processor using up to 1 GB of RAM. 4 GB limit on database size. No limit on the number of users. Impressive integration with .NET and Visual Studio. Can be enabled as a spatial DBMS by downloading the free Manifold Spatial Extender for SQL Server from the Updates page on the manifold.net web site.

New users skimming this documentation for the first time might mistakenly think that to function correctly Manifold requires installation of a DBMS package. That is not the case. All editions of Manifold include within Manifold itself very sophisticated and powerful DBMS capabilities so that Manifold may be used without any need to install additional DBMS software.

However, in addition to Manifold's internal DBMS capabilities, Manifold can also utilize external DBMS packages like the three mentioned in this topic to provide additional benefits to users. Especially in an enterprise setting, the use of a standard DBMS can provide additional power and capacity, enhanced collaboration with many simultaneous users and greater interoperability with applications already working with that DBMS.

Although the use of a major DBMS package has traditionally been thought of as something of interest only to enterprise users, because the Express versions of the major DBMS packages are so easy to install and because the Manifold interfaces for working with such DBMS packages are so much simpler than was previously the case with older software, many individual users or small groups have found that using a DBMS as a centralized geospatial data warehouse makes sense for them even though only one person is doing GIS.

### Three World-Class Choices

Recent years have seen the emergence of freely distributed "Express" editions introduced by the "Big 3" database vendors: IBM, Microsoft and Oracle.

These editions typically include the main features of the vendor's big-time "enterprise" database product but have been limited in some way. The idea is to make it possible for independent software vendors (ISVs) like Manifold to incorporate usage of such database products so that customers can get involved with the database product, fall in love and then ultimately buy the full-fledged, enterprise version of the database product as their needs grow.

It’s a highly effective strategy, as Microsoft proved by giving away free versions of the "Jet" database engine used in Access for use by ISVs. The result was tens of thousands of software applications that all standardized on the database engine used by Access. Unsurprisingly, when it came time for users of such applications to buy a DBMS package, they mostly bought Access and competitors were extinguished.

When Microsoft entered enterprise DBMS markets with SQL Server, they tried to repeat that strategy by offering the SQL Server database engine in a free, limited edition of SQL Server. Both Oracle and IBM countered that strategy by introducing free Express editions of their own products that raised the stakes on the original SQL Server Desktop Edition.
In a gutsy move, Oracle incorporated a significant amount of its elite spatial technology within Oracle Express. IBM entered the fray by introducing an Express edition of DB2 that has no limits on the number of users that may use the product or on the size of databases. Microsoft pushed forward with an enhanced free edition called SQL Server Express Edition that features especially tight integration with .NET and Visual Studio. The result is that all three Express editions deliver products of astonishing range, power and quality at no additional charge.

In the summer of 2007 IBM made available at no charge the IBM Spatial Extender for DB2 for use with DB2 Express-C, and the Manifold Spatial Extender for SQL Server became available at no charge for Manifold licensees. These developments migrated all three Express editions into competition as spatial DBMS packages as well.

The traditional way to limit a free DBMS version is to limit the number of processors it can use, the size of the database it can host and the amount of RAM it can use. This allows free, extensive use of the package in smaller applications while reserving the profitable, higher end applications for the paid version that has no such limits. Oracle and SQL Server both limit their Express editions to only one processor, one gigabyte of RAM and databases no larger than four gigabytes. IBM limits DB2 Express-C to two processors, four gigabytes of RAM and no limits on the size of the database.

Whichever DBMS Manifold users choose, the availability of three of the best enterprise-class DBMS packages ever created for free download is a testament to the extraordinary competitiveness, technical skill and confidence of these three DBMS vendors. Manifold users can install these database servers to create Enterprise servers for use with Enterprise Edition or as centralized data sources to save drawings that will be accessed by other users.

**No Technical Support**

Important: Manifold.net does not provide any support whatsoever for installation, administration, management, configuration or use of the DBMS products provided by IBM, Microsoft or Oracle.

If you download one of the database packages mentioned above, please keep in mind they are not supported by manifold.net and usually they are not supported by the DBMS vendors, either. If you want a supported DBMS installation, you must purchase a supported DBMS product from one of the DBMS vendors or you must purchase support products, if such are available, for the free DBMS product you are using from the vendor of that DBMS product. Manifold.net does not sell support products for the free DBMS products cited in this documentation.

Please do not waste technical support tokens by sending in tokens for questions about the free DBMS products. Doing so will simply waste a token to no good purpose. See the Technical Support topic for general information on tech support and using tech support tokens.

Developer level support incidents may be used for questions about Enterprise Edition features used with these three database products, but only for the actual Enterprise dialog. For example a question about connecting with Enterprise Edition via Database Console to an Oracle data source will result in an answer limited to the use of the Manifold dialog, such as explaining the purpose of the Server, User Name and Password boxes. Support does not extend to explaining how you can configure users in the DBMS, how to determine what server name is being used or should be used or how to determine whether a given user has connectivity to a given data source.

The DBMS products provided for free by major vendors are each supported by a vast industry of training products. Hundreds of books, for example, have been written on SQL Server alone, ranging from texts aimed at "dummies" to those covering the most sophisticated uses imaginable. Users intending to take advantage of these DBMS products should acquire and utilize the appropriate educational materials. It's not that difficult and many resources are standing by to help people learn and use these wonderful products.

IBM, Microsoft and Oracle are each doing a really extraordinary thing by providing free access to their incredibly elegant and powerful DBMS packages. Each vendor obviously has good business reasons for doing so, in the hope that later sales of full versions will justify the expenses of offering Express editions at no charge, but that only works if the DBMS vendors are not overwhelmed by uncooperative folks seeking free support for the free Express editions. Let's all work together not to ruin a good thing.

**See Also**

Database Administrator Edition
Database Console
Database Installations
Enterprise Edition

IBM DB2 Express-C Edition
Oracle Express Edition
SQL Server Express Edition
IBM DB2 Express-C Edition
At the present writing, IBM provides for free a complete, downloadable distribution of IBM® DB2® Express-C Edition. IBM DB2 Express-C provides the power of IBM DB2 to Manifold System users at no additional charge.

IBM DB2-Express C may be used by Manifold Enterprise Edition users to configure Enterprise Servers for use with Enterprise Edition or to store drawings using geometry data types as well as tables. See the Geometry in Tables topic.

Introduction
IBM DB2 Express-C Edition is derived from the same engine upon which IBM DB2 is built, so it shares the very high performance of IBM's flagship DBMS products. When delivered in IBM DB2 Express-C form, IBM has limited the database engine in two key ways:

- IBM DB2 Express-C may be installed on a multiple CPU machine, but it will execute (run) only on up to two processors or up to two processor cores if multi-core processors are used.
- IBM DB2 Express-C may be installed on a server with any amount of memory, but will use only up to 4 GB of available RAM memory.

Other than these limitations IBM DB2 Express-C provides essentially the full power and breadth of IBM DB2 capabilities. In particular, there is no limit on the size of databases and no limit on the number of simultaneous users.

Manifold System documents use of IBM DB2 Express-C within Manifold applications for several purposes:

- All Manifold System users can use DB2 Express-C to create high performance databases in which tables, drawings, images and surfaces may be stored. Those tables, drawings, images and surfaces can then be imported or linked into Manifold projects. Users may also write scripts that exploit the power of DB2 for the creation of databases and tables that may be used by their projects.
- All Manifold System users can store drawings within DB2 databases for concurrent, multi-user editing of drawings. If we have Manifold Database Administrator Edition we can configure DB2 storage using the Administrator Console to enable user-friendly features such as friendly names, formatting storage for drawings and pre-set import and link options.
- Manifold Enterprise Edition users can create Enterprise servers within DB2 databases hosted by a DB2 Express-C server. Because there is no limit to the size of databases or the number of users permitted to a DB2 Express-C installation, it is a good choice for Enterprise Server applications.
- Manifold Enterprise Edition users can create a spatial index for drawings within DB2 Express-C databases to confer true spatial DBMS capability unto DB2 Express-C. Once Enterprise Edition has been used to create a spatial index within a given DB2 Express-C database, users of any Manifold System edition can take advantage of that spatial index, for example, to link objects from drawings within a given area of interest.
- IBM offers a Spatial Extender for DB2 that provides IBM's own spatial DBMS capability for DB2. As of the summer of 2007, the Spatial Extender became available as a free download for DB2 Express-C. Manifold supports native DB2 / Spatial Extender functionality for those users who are working with Spatial Extender. Although Manifold allows using geometry types from other vendors (including Manifold's own geometry type) together with generic spatial indices established by Manifold, if the DBMS vendor has their own spatial types it makes sense to use them. Therefore, if you are working with DB2 / Spatial Extender manifold.net strongly recommends using native DB2 geometry types and not generic spatial indices or some other geometry type for drawings.

Because IBM DB2 Express-C is the same as IBM DB2 except for the processor / RAM limitations, if a Manifold application outgrows these limits it is very easy to scale upward by installing a full IBM DB2 license. Usually when an application grows so large that it outgrows IBM DB2 Express-C limits it is sufficiently important to merit the additional expense of procuring a full IBM DB2 license.

Installation and Configuration
The installation package currently downloadable from IBM installs IBM DB2 Express-C and configures it to start automatically on system startup.

IBM has distributed several versions of IBM DB2 Express-C. IBM DB2 Express-C Edition is used in this documentation since it is widely used. Consult the applicable IBM website for information on newer versions that may be available.
**Limitations**

IBM DB2 Express-C Edition may be used for Enterprise Edition Enterprise Servers without any technical limitations. However, two small usage limitations apply when DB2 Express-C is used for storing geometry in tables in the database to support subsequent concurrent, multiuser editing. When a drawing is linked from a DB2 Express-C data source the following limitations apply:

- Adding a new object and immediately editing it without refreshing the drawing creates an editing conflict.
- Adding a new object and immediately deleting it without refreshing the drawing will fail.

Therefore, when ever editing drawings linked from a DB2 Express-C data source make sure to refresh the drawing after adding a new object before attempting to edit that object or to delete that object.

**No Technical Support**

*Important:* Manifold.net does not provide any support whatsoever for installation, administration, management, configuration or use of the DBMS products provided by IBM. See the Database Installations topic for details.

**See Also**

Database Administrator Edition  
Database Console  
Database Installations  
Enterprise Edition

**IBM Links**

Links to websites change frequently. The links below have been provided to give an idea of what information is typically available via the vendor's websites.


The following links and descriptions have been provided by IBM:

**Article: DB2 Express-C, the developer-friendly alternative:**

[http://www.ibm.com/cgi-in/software/track0.cgi?i=53013&c=74646&o=1&ef=T&cn=7600831](http://www.ibm.com/cgi-in/software/track0.cgi?i=53013&c=74646&o=1&ef=T&cn=7600831)

This article shows how you can get started quickly using DB2 Express-C for all of your applications, and also provides an automation and tuning scenario to optimize your application.

**developerWorks: Resources for DB2 Express-C:**

[http://www.ibm.com/cgi-in/software/track0.cgi?i=53014&c=74646&o=2&ef=T&cn=7600831](http://www.ibm.com/cgi-in/software/track0.cgi?i=53014&c=74646&o=2&ef=T&cn=7600831)

These developerWorks pages feature technical and support information, and help you find the information you need to start using the DB2 Express-C features and functions. The pages also include non-technical information for your extended development or management teams.

**DB2 Technical Library:**

[http://www.ibm.com/cgi-in/software/track0.cgi?i=53015&c=74646&o=3&ef=T&cn=7600831](http://www.ibm.com/cgi-in/software/track0.cgi?i=53015&c=74646&o=3&ef=T&cn=7600831)
The developerWorks information team has recently redesigned the DB2 Technical Library search engine and put the most popular collections of information, articles, product documentation, Redbooks, and tutorials, in an easy-to-use format. Use this tool to locate the DB2 information you need.

Media: DB2 Express-C Discovery Kit (DVD):

http://www.ibm.com/cgi-in/software/track0.cgi?i=53016&c=74646&o=4&ef=T&cn=7600831

Prefer to get your product code and content on media? DB2 Express-C code images and related resources are available on DVD for Windows and Linux platforms. Request a no-charge copy for yourself.

DB2 Redbook Library:

http://www.ibm.com/cgi-bin/software/track0.cgi?i=47501&c=74648&o=1&ef=T&cn=7600833

IBM Redbooks are written by leading IBM technical experts. Small teams write the Redbooks during four- to six-week residencies. Sample these professional-grade publications online at no charge.

Porting to DB2 Universal Database:

http://www.ibm.com/cgi-bin/software/track0.cgi?i=53017&c=74648&o=2&ef=T&cn=7600833

The developerWorks DB2 Universal Database porting site gives you the information you need to port an application and its data from other database management systems to DB2 UDB.

How to buy IBM software products:

http://www.ibm.com/cgi-in/software/track0.cgi?i=51964&c=74646&o=5&ef=T&cn=7600831

IBM DB2 Universal Database Express-C is a no-charge product which is released without formal IBM support. If you decide you need 24/7 support from IBM, you must upgrade to the full version of DB2 Express, or other qualified DB2 products. There are a number of ways to buy IBM software. Please choose the method that best fits your needs.
Oracle Express Edition

At the time of writing, Oracle provides at no charge a downloaded distribution of Express versions of Oracle. Because many people have downloaded it in the past and now use it, this topic uses Oracle® 10g Express Edition. More recent Express editions of Oracle software will probably follow similar procedures. Oracle Express provides the power of Oracle to Manifold System users at no additional charge. Oracle Express Edition is often referred to as Oracle XE for short.

Oracle XE may be used by Manifold Enterprise Edition users to store Manifold components. It may also be used to store drawings using geometry data types as well as tables. Oracle XE includes Oracle's SDO_GEOMETRY technology from Oracle Spatial, which provides superior performance and a range of extra features such as the ability to link only part of a drawing in the desired area of interest in Manifold Enterprise Edition. Oracle XE does not include support for GeoRaster technology, so it can not be used to store images or surfaces in GeoRaster form. See the Oracle Spatial Facilities topic for additional information.

Introduction

Oracle Express Edition is derived from the same engine upon which Oracle Enterprise Edition is built, so it shares the very high performance of Oracle's flagship DBMS products. When delivered in Oracle Express form, Oracle has limited the database engine in several ways:

- User data cannot exceed 4 gigabytes in size (in addition to Oracle system data).
- Each computer can host only one instance of Oracle Express.
- Oracle Express may be installed on a multiple CPU machine, but it will execute (run) only on a single processor or a single processor core if multi-core processors are used.
- Oracle Express may be installed on a server with any amount of memory, but will use only up to 1 GB of available RAM memory.

Other than these limitations Oracle Express provides essentially the full power and breadth of Oracle capabilities.

Manifold documents use of Oracle Express within Manifold applications for several purposes:

- All Manifold System users can use Oracle Express to create high performance databases in which tables, drawings, images and surfaces may be stored. Those tables, drawings, images and surfaces can then be imported or linked into Manifold projects. Users may also write scripts that exploit the power of Oracle for the creation of databases and tables that may be used by their projects.
- All Manifold System users can store drawings within Oracle databases for concurrent, multi-user editing of drawings. Users of Manifold Enterprise Edition can use native Oracle connections for increased performance, use native Oracle data types such as SDO_GEOMETRY, restrict objects in a linked drawing to a desired area of interest, automatically create spatial indices and triggers for increased performance and employ automatic handling of routine database administration tasks when exporting drawings as well as other benefits. If we have Manifold Database Administrator Edition we can configure Oracle storage using the Administrator Console to enable user-friendly features such as friendly names, formatting storage for drawings and pre-set import and link options.
- Manifold Enterprise Edition users can create Enterprise servers within Oracle databases hosted by a Oracle Express server.
- Manifold provides a set of generic spatial DBMS facilities that are normally used to confer spatial DBMS capability to DBMS products that do not have their own spatial capabilities. Because Oracle is a very capable DBMS in addition to providing Oracle's own spatial technology, these generic facilities work with Oracle as well. However, it does not always make sense to use them. For example, in addition to the use of Oracle's native spatial DBMS capabilities, Manifold Enterprise Edition users can create a generic spatial index for drawings within Oracle Express databases. Given the universal presence of Oracle's own spatial technology within Oracle Express this does not make sense and is not recommended.
- A full license for Oracle Spatial (not just Oracle Express) will include Oracle's GeoRaster technology for storing images. If you are using a GeoRaster capable Oracle product it makes sense to use Oracle's own storage technology and not Manifold's generic spatial DBMS storage technology for images and surfaces. However, those users who do not have access to Oracle GeoRaster may want to use Manifold's generic facilities to store images and surfaces within Oracle Express.

Because Oracle Express is the same as Oracle except for the user data size limitation and processor / RAM limitations, if a Manifold application outgrows these limits it is very easy to scale upward by installing Oracle Enterprise Edition or other Oracle product. Usually when an application grows so large that it outgrows Oracle Express limits it is sufficiently important to merit the additional expense of procuring Oracle Enterprise Edition.
**Installation and Configuration**

The installation package currently downloadable from Oracle installs Oracle Express and configures it to start automatically on system startup. See the Installing Oracle topic for details.

**Further Reading**

Because Oracle Express is, in fact, Oracle there is a massive amount of reading material available to users who wish to learn more. In general, any documentation for Oracle will apply to Oracle Express so long as one keeps in mind the above differences between Oracle and Oracle Express.

When searching the Oracle.com site for additional information, search for Oracle Express Edition. Useful links that were current at the time this documentation was written include the following links.

**A Security Note**

When Oracle Express is installed it uses Windows authentication by default. It is strongly recommended that the user work with Oracle Express using Windows authentication.

**No Technical Support**

**Important:** Manifold.net does not provide any support whatsoever for installation, administration, management, configuration or use of the DBMS products provided by Oracle. See the Database Installations topic for details.

**See Also**

- Database Administrator Edition
- Database Console
- Database Installations
- Enterprise Edition
- Oracle Spatial Facilities
SQL Server Express Edition

At the time of writing, Microsoft provides at no charge a downloaded distribution of Express versions of SQL Server. Because it has been widely used in the past, this documentation uses Microsoft® SQL Server™ 2005 Express Edition as an example. Procedures for more recent versions of SQL Server Express are similar.

Microsoft also available by download the newest editions of SQL Server, such as Microsoft® SQL Server™ 2008 Express Edition. SQL Server 2008 has built-in spatial capabilities. SQL Server 2005 does not, but can be used as a spatial DBMS using the free Manifold Spatial Extender for SQL Server.

Introduction

SQL Server 2005 Express Edition is derived from the same engine upon which Microsoft SQL Server 2005 is built. It is a newer and, in many key ways, better alternative to the SQL Server 2000 Desktop Engine (known as MSDE) distributed in earlier Manifold releases.

When delivered in SQL Server Express form, Microsoft has limited SQL Server in several ways:

- A database cannot exceed 4 gigabytes in size.
- SQL Server Express may be installed on a multiple CPU machine, but it will execute (run) only on a single processor or a single processor core if multi-core processors are used.
- SQL Server Express may be installed on a server with any amount of memory, but will use only up to 1 GB of available RAM memory.

Other than these limitations SQL Server Express provides virtually the full power and breadth of SQL Server capabilities. Although MSDE was limited to no more than five users, SQL Server Express no longer has such an artificial throttle on performance. Within the limits of processor, RAM and maximum database size, SQL Server Express always runs at full speed.

Manifold documents use of SQL Server Express within Manifold applications for several purposes:

- All Manifold System users can use SQL Server Express to create high performance databases in which tables, drawings, images and surfaces may be stored. Those tables, drawings, images and surfaces can then be imported or linked into Manifold projects. Users may also write scripts that exploit the power of SQL Server for the creation of databases and tables that may be used by their projects.
- All Manifold System users can store drawings within SQL Server databases for concurrent, multi-user editing of drawings. If we have Manifold Database Administrator Edition we can configure SQL Server storage using the Administrator Console to enable user-friendly features such as friendly names, formatting storage for drawings and pre-set import and link options.
- Manifold Enterprise Edition users can create Enterprise servers within SQL Server databases hosted by a SQL Server Express server.
- Manifold Enterprise Edition users can create a spatial index for drawings within SQL Server Express databases to confer true spatial DBMS capability unto SQL Server Express. Once Enterprise Edition has been used to create a spatial index within a given SQL Server Express database, users of any Manifold System edition can take advantage of that spatial index, for example, to link objects from drawings within a given area of interest.
- Manifold users may download at no charge the Manifold Spatial Extender for SQL Server that provides high performance spatial indices for SQL Server. Users working with SQL Server 2005 Express should always download and install the Manifold spatial extender.
- Microsoft provides native spatial DBMS capability in SQL Server 2008. Manifold supports native SQL Server 2008 spatial functionality. Although Manifold generic spatial indices and the Manifold Spatial Extender for SQL Server will work with SQL Server 2008 spatial, if you have access to SQL Server 2008 spatial you should use that in preference to either generic spatial indices or the Manifold spatial extender.

Because SQL Server Express is the same as SQL Server except for the processor and RAM limitations and four-gigabyte database limit, if a Manifold application outgrows these limits it is very easy to scale upward by installing SQL Server. Usually when an application grows so large that it outgrows SQL Server Express limits the application is sufficiently important to merit the additional expense of procuring SQL Server.

This topic introduces SQL Server Express in the context of simple, default use with Enterprise Edition. If you do not have Enterprise Edition you will not have Enterprise features such as the Server Console. For more sophisticated configuration and usage of SQL Server Express, see the Microsoft documentation for SQL Server and SQL Server Express.
Installation and Configuration

The installation package currently downloadable from Microsoft installs SQL Server Express and configures it to start automatically on system startup. Install SQL Server Express and restart the system. SQL Server Express will appear as a SQL Server installation on the system.

Installing SQL Server Express Edition requires that we first install .NET Framework 2.0. The latest version of .NET Framework 2.0 available at the time of publication may be downloaded from Microsoft.

Installing SQL Server Express Edition:

1. In Windows Explorer, navigate into the folder into which you downloaded the installation files for SQL Server Express Edition and double-click the .EXE file.
2. In the End User License Agreement step of the installation, check "I accept the licensing terms and conditions" and click Next.
3. In the Installing Prerequisites step, click Install, wait until all prerequisite components are installed, and then click Next.
4. In the Welcome to the Microsoft SQL Server Installation Wizard step, click Next.
5. In the System Configuration Check step, inspect all generated warnings and click Next.
6. In the Registration Information step, click Next.
7. In the Feature Selection step, select the desired features and click Next. Default settings work OK.
8. In the Authentication Mode step, select Windows Authentication Mode and click Next. It is recommended to always use Windows Authentication Mode, unless there is a well-understood, significant reason not to.
9. In the Error and Usage Report Settings step, choose to automatically send error reports or feature use data to Microsoft, if you so desire, and click Next.
10. In the Ready to Install step, click Install.
11. In the Setup Progress step, wait until the installation completes, then click Next.

After the installation completes, you will be able to connect to the newly created instance of SQL Server using an administrative login from the machine on which SQL Server has been installed. You may want to configure the instance so it can be accessed by users who do not have administrative rights and/or users working on different machines.

Users planning on using SQL Server 2005 Express as a spatial DBMS should download and install the Manifold Spatial Extender for SQL Server onto the SQL Server machine.

While SQL Server configuration questions are outside of the scope of this Help topic, the following section provides an example of configuring an instance of SQL Server Express for use by ordinary users of a particular domain over a network. The following assumes execution right after completing the installation as shown above, using the same account which has administrative rights:

Allowing other machines to see an instance of SQL Server

1. Press the Start menu button and launch Microsoft SQL Server 2005 - Configuration Tools - SQL Server Configuration Manager.
2. In the tree on the left, select SQL Server 2005 Configuration Manager (Local) - SQL Server 2005 Network Configuration - Protocols for SQLEXPRESS. In the pane on the right, right click the TCP/IP item and select Enable. Click OK to dismiss the warning that the changes will not take effect until the SQL Server service is stopped and restarted.
3. In the tree on the left, select SQL Server 2005 Configuration Manager (Local) - SQL Native Client Configuration - Client Protocols. In the tree on the right, make sure the following items are enabled: Shared Memory, TCP/IP, Named Pipes. If any of the items is not enabled, right click it and select Enable.
4. In the tree on the left, select SQL Server Configuration Manager (Local) - SQL Server 2005 Services. In the pane on the right, right click the SQL Server Browser item and select Properties. In the SQL Server Browser Properties dialog, switch to the Service tab, make sure the Start Mode option is set to Automatic, then click OK. Right click the SQL Server Browser item again and select Start. Right click the SQL Server (SQLEXPRESS) item and select Restart.
5. Close SQL Server Configuration Manager.

6. If you have Windows Firewall turned on, turn it off. Alternatively, use the SQL Server Configuration Manager application to configure SQL Server to use specific TCP ports, then launch Control Panel - Windows Firewall, select the **Exceptions** tab, and specify the configured ports as exceptions.

**Allowing other users to connect to and use an instance of SQL Server**

7. Press the **Start** menu button and launch **Command Prompt**.

8. Execute the following command line:

```
SQLCMD -S (local)\SQLEXPRESS -E
```

9. Execute the following commands to create a new database and switch to it:

```
CREATE DATABASE test
GO
USE test
GO
```

10. Execute the following commands to allow users of the domain **MYDOMAIN** to connect to SQL Server and access the new database:

```
CREATE LOGIN [MYDOMAIN\Domain Users] FROM WINDOWS
GO
EXEC sp_grantdbaccess 'MYDOMAIN\Domain Users'
GO
```

11. Execute the following commands to assign one of the users of the domain **MYDOMAIN** as the owner of the new database:

```
EXEC sp_changedbowner 'MYDOMAIN\John'
GO
```

12. Execute the following commands to allow another user of the domain **MYDOMAIN** to write data to the new database:

```
EXEC sp_addrolemember 'db_ddladmin', 'MYDOMAIN\Jane'
EXEC sp_addrolemember 'db_datawriter', 'MYDOMAIN\Jane'
GO
```

13. Execute the following commands to allow all other users of the domain **MYDOMAIN** to read data from the new database:

```
EXEC sp_addrolemember 'db_datareader', 'MYDOMAIN\Domain Users'
GO
```

14. Execute the following command to exit SQL Server console:

```
EXIT
```

After following the above steps it is easy to test that a SQL Server Express installation is functional by using the Database Console in Manifold to quickly connect to SQL Server.

It is possible to connect using either ODBC, OLE DB or ADO .NET. We recommend using OLE DB for read-write connections and for all spatial DBMS work. We also recommend installing SQL Server Native Client on all client
machinges and then using the SQL Native Client provider. If we haven’t installed the SQL Server Native Client we can connect to SQL Server using the default OLE DB Provider for SQL Server.

Connecting to a SQL Server Express Installation:

1. Launch Manifold and launch the Tools - Database Console command.
2. Press the [...] browse button to the right of the Data source box.
3. Follow the instructions in the Create a Data Source for SQL Server Express example section of the Data Source Dialog topic to create a data source for the desired SQL Server Express installation in the Data Source dialog. Double click on that data source to connect to it.

Once a data source for a SQL Server Express installation has been created the Data Source dialog will remember it. It will be available for subsequent uses with Database Console as well as with Administrator Console (if we have Database Administrator Edition installed).

Configuring SQL Server Express as an Enterprise Server

This section is for Manifold Enterprise Edition users. As discussed in the Data Storage Strategies topic, Enterprise Edition allows a simplified “shared” storage strategy that combines the benefits of file-based storage with those of full scale DBMS storage.

After installing SQL Server Express we can configure the SQL Server installation for use as an Enterprise Server.

To read (get) components shared on an Enterprise server, a user account must be allowed to connect to SQL Server, access the database, and be allowed to read data in the MFD_ROOT table that lists available components and data in the data tables for components stored in the server.

To read and write (get, check out, check in, undo check out, share, delete) components shared on an Enterprise server, a user account must be allowed to do all of the above and also to alter data in the MFD_ROOT table and in the data tables, as well as to create and drop tables.

If we install SQL Server Express as given above and if we have followed steps 1 through 14 above given in the section titled Allowing other machines to see an instance of SQL Server then the user accounts MYDOMAIN\John and MYDOMAIN\Jane will be able to use the database Test as an Enterprise server in read/write mode, and all other users of MYDOMAIN will be able to use it as an Enterprise server in read-only mode.

If we had installed SQL Server Express on a machine that is not part of any domain we could get the same result (John and Jane having read/write and everyone else read-only rights) by altering the procedure given above so that commands used in SQLCMD in steps 9 through 14 use BUILTIN instead of the domain name MYDOMAIN.

Using Database Console for Configuration

While it is certainly possible to configure SQL Server Express by using SQLCMD from the Windows Command Prompt on the machine on which it has been installed as illustrated above, as a practical matter it is frequently more convenient to configure it remotely by using Database Console.

Instead of logging in to the machine running an instance of SQL Server Express and launching SQLCMD we could log in to one of the machines that can connect to the server using a user account that has sufficient permissions on the instance of SQL Server Express (for example, a member of the sysadmin server role), launch Manifold, connect to SQL Server Express using Database Console and execute commands by typing them in the query pane and clicking the Run toolbar button. When executing a command in the query pane we do not have to use GO to signal the server that it is time to execute the command, since the Run button does that for us.

A quick and dirty way to allow all users of a particular domain to use a particular database as an Enterprise server in read/write mode is to make them all members of the built-in db_owner role for that database.

To do this via SQLCMD, we can execute the following commands (assuming the configuration above):

\texttt{USE test}
Here are the equivalent commands for an instance of SQL Server on a machine that is not on a domain and with no configuration done previously via SQLCMD:

```
CREATE DATABASE test
GO
USE Test
GO
CREATE LOGIN [BUILTIN\Users] FROM WINDOWS
EXEC sp_grantdbaccess 'BUILTIN\Users'
EXEC sp_addrolemember 'db_owner', 'BUILTIN\Users'
GO
```

Here is how we would do the above using Database Console instead of SQLCMD:

```
CREATE DATABASE test

(click the Run button)

USE Test

(click the Run button)

CREATE LOGIN [BUILTIN\Users] FROM WINDOWS
EXEC sp_grantdbaccess 'BUILTIN\Users'
EXEC sp_addrolemember 'db_owner', 'BUILTIN\Users'

(click the Run button)
```

To remove a database from SQL Server Express, in SQLCMD use:

```
DROP DATABASE test
GO
```

Use with Enterprise Edition

Once SQL Server Express has been installed and a database has been created on the server we can use it as an Enterprise server. We can create an ODBC or OLE DB data source to connect to it as set forth in the Creating an Enterprise Server topic, or we can use shorthand Enterprise Edition syntax provided for a simplified connection to SQL Server databases.

Enterprise dialogs allow a simplified connection string syntax for SQL Server databases as an optional alternative to full ODBC syntax. \"\systemname\" connects to the default database on a given system. \"\systemname:database\" connects to the specified database on the given system. The connection established with a simplified connection string that uses Windows integrated security (suitable for users working in Windows Server 2003, Windows XP and Windows 2000). Using simplified connection strings is a fast way to connect to SQL Server or to SQL Server Express using default options without dealing with the ODBC dialogs.

To connect to a non-default instance of SQL Server Express, expand systemname into systemname\instancename. Thus, connecting to \"\mysystemsqlexpress:test\" will connect to a machine named mysystem, an instance of SQL Server Express on that machine named sqlexpress, and a database on that instance named test.
For example, if we have installed SQL Server on a system called PROJECTS and we have created a database called hydrography to use as our Enterprise server database, we could connect in the Server Console by entering \projects:hydrography into the Server box of the Server Console and then pressing the Refresh button.

Sharing a Component into a SQL Server Express Enterprise server

1. Launch Manifold and open the project containing the component of interest.
2. Right click onto the component and choose Share.
3. Suppose the system hosting the SQL Server Express server is called PROJECTS and the database we created is called manifold. In the Share on box enter \projects:manifold and press OK.

The Enterprise server will be created in the manifold database and the component will be shared into that server. Thereafter, whenever we want to work with that Enterprise server we can open the Server Console and in the Server box enter \projects:manifold to work with the contents of that SQL Server Express server. Of course, if we work with this server on a regular basis Manifold will remember the last-used entries in the Server box for our convenience.

Using an Enterprise server with SQL Server Express

1. Launch Manifold and open a project.
2. Choose Tools - Server Console to launch the Server Console dialog that allows browsing Enterprise servers.
3. In the Server box, the data source connection string or data source name that was last used will appear. To see the contents of any Enterprise server, enter the data source name for that server (such as Example Enterprise server) into the Server box. Press the Refresh button at any time to see the updated contents of that server if you think there have been changes. When working with SQL Server Enterprise servers we can use simplified syntax, as in \systemname:database
4. Click on the component desired and the toolbar buttons to Import or Link will be enabled.
5. Press Import to import the component into the project. This fetches a copy of the component from the Enterprise server into local project storage. Press Link to link the component into the project, leaving it stored within the Enterprise server and under control of the Enterprise server.

Example

Suppose we've just installed SQL Server Express on a machine called PROJECTS. We will create a database called counties and we will grant permissions to all users in our domain to access the database. Our Windows domain is called GISDEPT.

1. Launch Manifold and open Tools - Database Console. The Database Console uses the Data Source dialog to specify connections.
   a. Click the [...] browse button to the right of the Data Source box to open the Data Source dialog, which is similar to the File - Open dialog. In the Files of type box choose OLE DB Data Sources (). This opens the Data Link Properties dialog.
   b. Follow the instructions in the Create a Data Source for SQL Server Express example section of the Data Source Dialog topic to create a data source for the desired SQL Server Express installation in the Data Source dialog, using the Microsoft OLE DB Provider for SQL Server and connecting to the PROJECTS machine. Double click on that data source to connect to it.
2. In the command pane (the lower pane in the Database Console) enter the following command and press the Execute:

   CREATE DATABASE counties

3. In the command pane enter the following four lines. After all four lines have been entered press the Execute button:

   USE counties
   EXEC sp_grantlogin 'GISDEPT\Domain Users'
   EXEC sp_grantdbaccess 'GISDEPT\Domain Users'
   EXEC sp_addrolemember 'db_owner', 'GISDEPT\Domain Users'
4. Close the Database Console. To populate the Enterprise server with components, right click on a component to be shared to the server and choose Share. In the Share dialog, enter the following into the Share on box:

\projects:counties

5. The Enterprise server is now fully operational and populated with the shared component. Use \projects:counties in the Server box of the Server Console to work with that Enterprise server.

Comments

Any machine can host an Enterprise server run on SQL Server Express databases. Although Microsoft refers to it as a "desktop" engine it is actually SQL Server in all its full power and elegance, excepting the limitations of CPU / RAM usage and four gigabytes per database. It has full networking capability as does SQL Server and will appear on the local network as a SQL Server installation. For most practical purposes of interest to Manifold users everything that may be done with SQL Server may be done with SQL Server Express as well.

If we install Enterprise Edition with SQL Server Express on a given machine, that machine can be both a client and a server to other machines on the network. For example, if we have a local area network with ten different machines we can have each of the ten be an Enterprise server for itself and for other machines on the network, even as the machines are simultaneously used as Manifold Enterprise Edition client workstations.

The short examples given above show one database created within a SQL Server Express server. We could create many databases within each SQL Server Express server and use them all simultaneously. Each database is used as a separate Enterprise server.

For example, suppose on the machine called PROJECTS we install a SQL Server Express server and then create databases called hydrography, boundaries, transportation and utilities. We could work with each of these in the Server Console as four different Enterprise servers using the shorthand connection syntax of \projects:hydrography, \projects:boundaries, \projects:transportation and \projects:utilities. We could include shared components from each of these in the same project.

Further Reading

Because SQL Server Express is, in fact, SQL Server there is a massive amount of reading material available to users who wish to learn more. In general, most documentation for SQL Server will apply to SQL Server Express so long as one keeps in mind the minor differences between SQL Server and SQL Server Express.

When searching the microsoft.com site for additional information, search for SQL Server 2008 Express Edition, which has superseded the earlier SQL Server 2005 product.

No Technical Support

Important: Manifold.net does not provide any support whatsoever for installation, administration, management, configuration or use of the DBMS products provided by Microsoft. See the Database Installations topic for details.

See Also

Database Administrator Edition
Database Console
Database Installations
Enterprise Edition
Drawings

Drawings

Drawings in Manifold are similar to drawings in a CAD system used to create blueprints. They are made up of points, lines or areas and are often referred to as vector drawings in other mapping programs or CAD systems.

Existing drawings may be imported into a Manifold project using the File - Import - Drawing command. New drawings may be created in Manifold and then exported into common drawing file formats using File - Export. See the Import and Export topic for more information on specific import dialogs used to import drawings from various GIS formats.

Creating a New Drawing

1. Open a project or create a new project with File - New.
2. Choose File - Create - Drawing to create a new drawing.

The File - Create - Drawing dialog takes the projection parameters for the new drawing from whatever window is active at the time the new drawing is created. If the project pane is active, the drawing will be created using the system default projection of Orthographic centered at the 0,0 world latitude/longitude origin. If a map window or other drawing window is active when the new drawing is created, the new drawing will be created using whatever projection is used by that active window. This context-sensitive setting of default projection parameters makes it much easier to create new drawings using projection parameters that are hassle-free by default.

Every drawing has a table associated with that drawing. Creating a drawing will automatically create the table for it.

Drawing Fundamentals

Drawings are fundamentally different from images in several ways:

- Drawings are made up of objects drawn using points, lines and areas that are defined by specific coordinates. In contrast, images have no objects, only a sea of pixels. The "objects" we see in an image appear to be there because our eyes and brains automatically match up pixels of similar colors to interpret that part of the image as being an object.

- The appearance of objects in a drawing is determined by the format of each object and not by the zoom level. A point in a drawing that is formatted to appear as a small circular dot will have the same size and appearance regardless of zoom level. Areas that are formatted so that they appear with a contrasting color on their border will have the same thickness of border line no matter how far one zooms in. Changes in zoom will simply show more or less of the object. In contrast, the appearance of images varies dramatically as one zooms in past the point where individual pixels become visible.

- Drawings use a coordinate system that specifies the precise geometric relationship between different parts of different objects. Drawings intended for use as geographic maps will use a geographic coordinate system so they automatically may be located on the surface of the Earth. In contrast, images have no intrinsic coordinate system except that which is implied by the row and column arrangement of pixels. Images that might be imported into Manifold are rarely bound to geographic coordinates and thus will usually require a quick georegistration before they can be used in geographic maps in a sensible way.

- Because drawings have distinct objects, a drawing can be linked with database information by attaching database records to objects in the drawing. In contrast, because there are no objects in images, databases cannot be attached to images. The only data in images are the color values of the pixels.

- Every drawing has a table associated with that drawing. Each row in the table corresponds to an object in the drawing. A drawing with three points in it will have three rows in its table, for example. The table will have at least one field, an object ID field, that is used to link each row in the table to an object in the drawing.
Points, Lines and Areas

Drawings are created from only three types of objects:

- Points
- Lines
- Areas

No matter how detailed or cleverly formatted a drawing may be, it is ultimately made up of only these three types of objects. Drawings do not have to contain all three types of objects. Many drawings contain only points, only lines or only areas.

**Points** are simply dots at a particular location. Points are formatted by default as small, round dots. Points can be shown as anything from a single pixel to a multicolored icon. In the illustration above, the three points to the left are shown as single pixels whereas the four points to the right are dots formatted with bright green background color.

**Lines** may appear to be straight lines or smooth curves. However, all lines are ultimately made up of straight-line segments drawn between the sequence of coordinates that define the line. Zooming far into even the most smoothly curved line will show that it is made up of straight-line segments. Some CAD and GIS systems call such lines "polylines." In Manifold, we simply say "lines."

**Areas** are like shapes cut out of cardboard. They are objects that actually have some "substance" in their inner region and are often used to show regions such as states or provinces in a map. The illustrations above show a few western states drawn using areas. At left we see the areas drawn in their normal position, and at right drawn separate from each other. The illustration at right has had "shadows" added in a simple way to provide a greater impression of substance, to promote the analogy with shapes cut out of cardboard.
**Historical Note:** Areas were called "polygons" and lines were called "arcs" in some older GIS systems. This terminology may still be used in some organizations that are still using legacy systems. See the Terminology in GIS essay for notes on the difference between older words and modern usage.

**Don’t Confuse Lines with Areas**

Areas are a new concept to users of simple drawing packages. Many graphics arts packages that allow the creation of vector drawings allow use of lines and points, but not areas. Other graphics packages, such as the drawing toolbar in Microsoft Word, uses “shapes” in a manner that is part way between line drawings and the use of areas in more sophisticated packages.

An important difference between areas and lines is that areas have inner substance, like a cardboard cutout, whereas even when lines are drawn in a way that makes a closed figure there is no "substance" inside the white space of the figure. A closed figure drawn using lines is like a figure made up of wire twisted into the desired shape. There’s nothing "inside" the figure.

If we redraw the Western states using lines instead of areas, we can see that when we pull apart the constituent lines there is nothing "inside" the states. It is as if we made a wireframe figure out of pieces of stiff wire twisted into the shape of the state boundaries. When we take the figure apart we see it is just bits and pieces of wire.

The distinction between drawings made up of lines and those made up of areas is important because the appearance of the objects depends on how they are formatted. Only the "substance" of the objects can be formatted. If we wish to draw Western states with different, smooth colors within their interior regions we should use areas to draw the states. We can’t color the interiors of a drawing of Western states made up of lines because there is no "interior" to color.

Imagine, for a moment, the drawing of states using lines as though it were made up of wires and seen in perspective. If we tried to paint the inside of the states with a paint bucket the paint would flow right through the empty spaces between the wires.
If we imagine the drawing made up of areas as though they were shapes cut out of cardboard we can pour paint onto the shape and have it color the interior of the area.

**Advantages of Lines**

It’s a good thing that figures drawn of lines have no interior substance. That makes it easy to use lines to draw linear features like roads in a layer that overlays other objects. One can then see the objects below through the nothingness between the lines.

The image above uses three line drawings stacked up above an image layer. One layer uses lines formatted as thick black lines, while the others use thinner lines. The illustration shows Palo Alto, California, with Highway 101 cutting across from left to right and San Francisquito Creek (the border between Santa Clara and San Mateo counties), shown in blue.

**Don’t Confuse Area Boundaries with Lines**

In a drawing created entirely from areas it could be difficult to see the "joints" between areas if they fit together perfectly.
Suppose we cut out the shapes of Western states from fine paper and fitted them together so carefully the seams between them could not be seen:

That's a less useful drawing than one that clearly shows the seams between the areas so that it's obvious where each area begins and ends:

By default, Manifold will use an area style to draw areas that draws the last pixel at their edges in a contrasting color. This makes it possible to see the seams between areas even when they are perfectly fitted together. When looking at areas drawn using this default area style, we might be tempted to think that we are looking at a map that includes a set of lines drawn in the shape of the areas; however, this is just a different area style. See the Areas and Boundary Lines topic for more discussion on this.

Using Lines to Duplicate Area Boundaries

Sometimes we will want to use both areas and lines to show the same thing, in order to take advantage of greater formatting flexibility.
It's easy in Manifold to create a set of lines that follow the boundaries of area objects using Transform - Boundaries. If we like, we can use this command to create lines that exactly follow the borders of our Western state areas. We can then stack these lines in a layer above the area layer. This gives us greater formatting flexibility than is possible using area styles alone. For example, we can make the lines layer using extra-thick lines in some boundaries to emphasize certain states.

Another reason to create a layer of lines that follow the borders of our state areas is to create a layer of lines that can be overlaid upon an image. Consider the following example:

Suppose we have an image of the US and a drawing that shows US states as areas.

If we stack the areas drawing in a layer above the image the substance of the areas will obscure the image. Although we might use a transparent area effect so that the image could show through the areas in this case we will use lines to get exactly the right visual effect desired.
To create the lines, we use **Boundaries** to create lines in the shape of the area boundaries.

We can create the new lines in a new drawing layer in our map. We can then stack the lines above the image to show boundaries in the image.

Note: this is a small image to fit into the Help documentation and so a very bright color has been used for the boundary lines. In "real life" it is often cooler to use more subdued colors to achieve a more modern look in one’s maps.

**Areas Can be Drawn in Opaque or Transparent Styles**

By default, areas are drawn using an opaque style in gray color. Objects below areas drawn in such style will not be visible. If you can’t see objects that you expect to be in your drawing, make sure that they are not on a layer that is below an opaque area. Move them above the area and they will become visible.

Areas may also be drawn using a variety of transparent hatch patterns or even in a translucent style (by changing the Layer Opacity of the layer they are in). In this case, they are like shapes cut out of transparent plastic or translucent paper.
Areas that overlap each other within the same layer will overlap above or below each other depending on the order in which they were created. When working with areas that overlap, it’s often helpful to use transparent area styles so that the region of overlap is clearly visible in all cases. See the Overlapping Objects and Transparent Area Styles topics for more information.

When drawings are used as layers in a map, the layers can be shown with different Layer Opacity so that different layers can be seen through those that are partially transparent. This is a fine way to show overlapping areas.

Suggestions

- Use lines to draw linear features like roads and streams.
- Use areas to draw regional features such as provinces or lakes.
- When formatting objects, keep in mind the visual analogy that lines are like wires hovering above the drawing and areas are like shapes cut out of cardboard.

Drawings and Maps

Drawings by themselves can only show points, lines or areas. To show other visual elements, such as labels or images, we use other Manifold components such as Images or Labels components. By combining drawings with images and labels layers in a map we can achieve exactly the visual effect desired.

Using the Layers Pane with Drawings

The Layers pane is used to control the appearance of drawings within drawing windows. The layers pane includes checkboxes for two system "layers" - a background color layer and a border layer that shows an enclosing box about the widest extents of objects in the drawing.

Drawings are shown using the checkerboard background Manifold uses to provide a backdrop for any transparent regions. The layers pane is shown to the right of the drawing window.
Checking the Background box in the layers pane will replace the checkerboard background with whatever is the default background color. The default setting for drawings is to show a white background.

Checking the Border box will draw a one-pixel border that represents the minimum enclosing box for all objects in the drawing. When working with drawings that include points, using the Border is a handy way to see if any very small objects exist far away and unnoticed from other objects, since the border will expand to include all objects.

Note that only maps can have true "layers" in Manifold in the sense that they can layer more than one component within the same map window. The border and background "layers" in the Layers pane for are not true layers even though they appear in the Layers pane in the same manner as do layers in maps. These are simply system controls that take advantage of the Layers pane as a conceptually convenient user interface.

Clicking off the Background in the layers pane can resolve visual ambiguities between areas that are white in color and regions that are bordered by lines. With Background on as in the illustration at left above it is not possible to immediately see solid areas that are colored in the same color as the background color. With Background off, white areas will stand out against the default checkerboard pattern.

We can change the background color in Tools - Options. See the Layers topic for an example of how changing background color can radically change the appearance of a drawing or map.

Layouts and the Layers Pane
If a drawing has any Layouts created they will appear as “layers” in the layers pane for that drawing. Checking the box for one of these print layout layers will cause a layout rectangle to appear in the drawing that shows the region covered by the layout.

In the illustration above the drawing has four layouts that show different parts of Mexico. Three of the layouts have been checked in the layers pane causing three layout preview rectangles to appear in the drawing.

Right clicking onto the hatched border of one of the layout rectangles in the drawing will cause a context menu to appear with controls based on that layout rectangle. For example, we can **Zoom** to a given layout rectangle, **Print** it or change its **Properties**. If a layout is empty (for example, if the layout scope is set to **selection** and nothing is selected in the parent component) zooming to the layout will do nothing.

Use **Tools - Options - Colors - Layout Rectangle** to change the color in which layout rectangles are shown. The default color is black.

**See Also**

- Format Toolbar
- View - Properties - Zooms
Drawing Window Menus and Controls

When we click onto a drawing window, Manifold will automatically change menus and toolbars so they contain those menus and controls that work with drawings. The following are commands that appear which are specific to drawings.

**Edit**

- **Cut**  Windows clipboard cut operation. Copy selected objects to the windows clipboard and delete them from the drawing.

- **Copy**  Windows clipboard copy operation. Copy selected objects to the Windows clipboard.

- **Paste**  Windows clipboard paste operation. Paste the contents of the Windows clipboard into the drawing deleting all previously selected objects. Creates new objects in the drawing.

- **Paste Append**  Paste the contents of the Windows clipboard into the drawing without deleting any selected objects. Creates new objects in the drawing.

- **Delete**  Permanently delete selected objects from the drawing.

- **Delete All**  Permanently delete all objects from the drawing.

- **Edit - Change Projection**  Used with components like drawings and images to re-project the component and thus permanently change the data.

- **Edit - Assign Projection**  Used with drawing, image, labels and surface windows to change the interpretation of the data. This is a specialized function that is applied only when manually specifying the projection of a component imported from a format that does not correctly store projection information. To change the native projection of any component other than a map, use the Edit - Change Projection command.

- **Select All**  Select all objects in the drawing.

- **Select None**  Deselect all objects in the drawing.

- **Select Inverse**  Invert selection: those objects that were selected will be deselected while those objects that were not selected will be selected.

- **Select Mode**  Choose the selection mode to be used for mouse selection:
  - **Replace**  - Any selection made with the mouse will replace the previous selection.
  - **Add**  - Any selection made with the mouse will be added to the previous selection.
  - **Subtract**  - Any selection made with the mouse will be subtracted from the previous selection.
  - **Invert**  - Any selection made with the mouse will be inverted with the previous selection.
  - **Intersect**  - Any selection made with the mouse will be intersected with the previous selection.

- **Select Objects**  Choose objects that may be selected by mouse selection:
  - **Areas**  - Any areas within the mouse selection will be selected.
  - **Lines**  - Any lines within the mouse selection will be selected.
selected.

**Points** - Any points within the mouse selection will be selected.

**Select by Type**
Select all objects of the currently enabled Select Object modes, for example, select all points or all lines. See the Selection in Drawings topic and the Edit - Select by Type topic.

**Snap To**
Enable / Disable Snap modes:
- **Graticule** - Snap to graticule intersections if a graticule is displayed.
- **Grid** - Snap to grid intersections if a grid is displayed.
- **Areas** - Snap to coordinates that define areas.
- **Lines** - Snap to coordinates that define lines.
- **Points** - Snap to points.
- **Segments** - Snap to locations within segments.
- **Selection** - Snap only to objects in the selection, if any.

**Instant Data**
Automatically pop up an input dialog after adding a new object to the drawing. Allows rapid data entry when creating new objects.

**Shared Edit**
Toggle shared edit mode.

**Go To**
Go to a particular object or location using a variety of options.

**View**
- **Back**
  Go back one view.
- **Forward**
  Go forward one view.
- **Zoom In / Out**
  Zoom in brings us closer to the drawing while zoom out moves us farther away.
- **Zoom To Fit**
  Zoom so that the drawing fits the existing window size.
- **Zoom To**
  Zoom to a specified zoom level.
- **Graticule**
  Show a graticule (latitude / longitude lines).
- **Grid**
  Show a grid that may be used to measure or to align editing and snap commands.
- **Legend**
  Display a legend showing formats.
- **North Arrow**
  Display a North arrow.
- **Scale Bar**
  Display a scale bar.
- **Structure**
  Display within the drawing window structural items for objects, including:
  - **Object Boxes** - The minimum enclosing rectangle for each object.
  - **Object Centroids** - The coordinate centroid for each object.
  - **Branch Boxes** - The minimum enclosing rectangle for each branch of multi-branched objects.
  - **Branch Centroids** - The coordinate centroid for each branch of a multi-branched object.
  - **Inflection Points** - The coordinates that define each
object.

**Segment Points** - The centroid of each line segment (from coordinate/inflection point to coordinate/inflection point) that defines each line or area.

Points, of course, have only one inflection point or segment point, located at the point's location itself. See the View - Structure topic for examples.

**Panes**

Call a dialog that allows displaying or hiding all panes.

**Full Screen**

Show the drawing over the entire monitor area.

**Refresh Data**

Update a linked drawing with the latest data from the controlling geocoded table.

**Refresh View**

Update the visual appearance of the drawing with any changes made since the last refresh. Enabled only if **Autorefresh View** is off and some change (such as a selection) has been made that might affect the visual appearance of the drawing.

**Autorefresh View**

Automatically update the drawing whenever any changes are made that might affect the visual appearance of the drawing. Turn this option off when working with very large drawings where there is no desire to take the time to show, say, a selection when it is made (repainting a very large drawing on every selection step might take longer than desired). When **Autorefresh View** is off, we can always cause a refresh on demand using the **Refresh View** command.

**Properties**

View the properties dialog for this drawing.

**Drawing**

**Open Table**

Open this drawing's table.

**Align**

Alignment toolbar commands that work with selected objects and the primary selected object (selected with a **CTRL-ALT** click):

- **Left** - Move objects so their left edges are aligned to the left edge of the primary selected object.
- **Top** - Move objects so their top edges are aligned to the top edge of the primary selected object.
- **Right** - Move objects so their right edges are aligned to the right edge of the primary selected object.
- **Bottom** - Move objects so their bottom edges are aligned to the bottom edge of the primary selected object.

**Center**

Alignment toolbar commands that work with selected objects (selected with a **CTRL-ALT** click):

- **Horizontally** - Move objects so that the X coordinate of the center of their bounding box coincides with that of the bounding box of all objects in the drawing.
- **Vertically** - Move objects so that the Y coordinate of the center of their bounding box coincides with that of the bounding box of all objects in the drawing.

**Space**

Alignment toolbar commands that work with selected objects (selected with a **CTRL-ALT** click):

- **Across** - Space objects evenly in horizontal direction across the bounding box of all objects in the drawing.
- **Down** - Space objects evenly in vertical direction across the bounding box of all objects in the drawing.
Resize Full Alignment toolbar commands that work with selected objects (selected with a CTRL-ALT click):
  Width - Increase width of objects to that of the bounding box of all objects in the drawing.
  Height - Increase height of objects to that of the bounding box of all objects in the drawing.
  Both - Increase both width and height of objects to those of the bounding box of all objects in the drawing.

Resize Same Alignment toolbar commands that work with selected objects (selected with a CTRL-ALT click):
  Width - Resize objects so that the width of each is the same as the width of the primary selected object.
  Width (Max) - Resize objects so that the width of each is the same as the width of the widest object.
  Width (Min) - Resize objects so that the width of each is the same as the width of the narrowest object.
  Height - Resize objects so that the height of each is the same as the height of the primary selected object.
  Height (Max) - Resize objects so that the height of each is the same as the height of the tallest object.
  Height (Min) - Resize objects so that the height of each is the same as the height of the shortest object.

Minicharts Create and format minicharts in this drawing.

  Color Color each adjacent area using a different color.
  Applies a thematic format based on a coloring field.

  Dissolve Merge area objects (dissolving away area borders) that have identical values in given data attribute fields.

  Districts Assign areas in the drawing to different districts by writing a district code to a specified data field.

  Orthogonalize Move coordinates to orthogonal grid positions. Used to trim insignificant digits.

  Segmentize Add redundant coordinates to increase the number of coordinates used to define the object. See the Segmentization topic for discussion.

  Simplify Remove coordinates that define the object to simplify the shape of the object.

  Spatial Overlay Transfer data between objects in drawings based upon their spatial relationships.

  Topology Overlay Modifies drawings using Identity, Intersect, Union and Update overlay functions using the areas in a source drawing to guide modifications in a target or results drawing.

  Relink Enabled for a linked drawing only. Relinks (reconnects) the linked drawing to another data source.

  Unlink Enabled for a linked drawing only. Converts the linked drawing to a local drawing with no connection to the originating data source. Linking and then unlinking a drawing is equivalent to importing it.

**Drawing Window Context Menus**

Context menus pop open when we right click on an item in Manifold. They are called “context” menus because what they show depends on the context of where the right click occurred.
Right Click onto an Object

- **Cut**: Windows clipboard cut operation. Copies object to the clipboard and delete it from the drawing.
- **Copy**: Windows clipboard copy operation. Copies object onto the clipboard without deleting it.
- **Paste**: Paste the contents of the Windows clipboard into the drawing. Creates new objects in the drawing, replacing any objects that were selected. New objects will be selected.
- **Paste Append**: Like **Paste**, but does not delete and replace any objects that were previously selected.
- **Delete**: Permanently delete this object.
- **Duplicate**: Duplicate the object. It is wise to immediately move the new object while it is selected. This avoids creating coincident objects that might later cause confusion.
- **Orthogonalize**: Move coordinates to orthogonal grid positions. Used to trim insignificant digits.
- **Segmentize**: Add redundant coordinates to increase the number of coordinates used to define the object. See the Segmentization topic for discussion.
- **Simplify**: Remove coordinates that define the object to simplify the shape of the object.
- **Center**: Pan the view to center this object in the window.
- **Zoom**: Zoom to fit this object.
- **Coordinates**: Launches the Object Coordinates dialog to allow direct editing of the coordinates that comprise the object.
- **Fields**: Launches the Object Fields dialog that shows any table fields associated with this object.

The **Cut**, **Copy** and **Delete** functions within the context menu of a drawing work slightly different than their equivalents found in the **Edit** menu. The **Edit** menu versions work with the selection. The context menu versions work on those that object that has been right clicked.

**Double Click Short Cuts**

Double-clicking on legends, north arrows and scale bars will open the **Properties** dialog for those items.

Double-clicking on an object in a drawing will launch your Internet browser using the contents of a **URL** field, if any, in the drawing's table for that object. If there is no **URL** field or if it is not of type "URL" or if it is empty the **Fields** dialog will be popped open.

**CTRL**-double-clicking on an object in a drawing will launch the **Fields** dialog.

**Minicharts**

Minicharts are small charts that hover above objects in a drawing. See the Minicharts topic for details.

**Frequently Used Panes**

Panes are launched from either the **View - Panes** menu or by **ALT-SHIFT** keyboard shortcuts listed in the **View - Panes** menu.
**Control Points**  Assign control points for geregistration.

**Info**  Display data attributes for objects in the drawing.

**Layers**  Turn background and border on/off, show coverage of any print layouts using this drawing.

**Selections**  Saved selections for this drawing.

**ViewBots**  Assign 'bots to watch this drawing and report an analysis.

**Views**  Save and return to particular pan and zoom view.
Selection in Drawings

Please read the main Selection topic before reading this topic. The Selection topic explains selection modes and the use of select object buttons.

Selection in drawings is the same as selection in drawing layers in maps. Since maps are the preferred user interface for most interactive work, selection is less often done directly within a drawing window. However, it works the same way in both cases.

Objects in drawings may be selected in several ways:

- Mouse moves such as Select Box or SHIFT Select Box
- Selection using mouse moves within any Tables linked to the drawing.
- Selection using the Query toolbar.
- Selection using SQL queries.
- Objects created by the Transform toolbar are automatically selected after they are created.
- Some transform toolbar operators (such as the Select Adjacent to operator) may select objects in the drawing.
- Scripts may select objects in drawings.
- A selection in a drawing or image can be used to create a selection in a drawing using the Transfer Selection command.

Selection using Mouse Moves

The key to selection using mouse moves is to choose the right selection mode and the desired combination of select object filter buttons. The select object filter buttons used with drawings are:

- **Select Areas** - Enable selection of areas by mouse selection tools.
- **Select Lines** - Enable selection of lines by mouse selection tools.
- **Select Points** - Enable selection of points by mouse selection tools.

Choose the type of items that are to be selected by pushing the selection filter buttons for the desired types. When working in drawing windows, the Select Pixels and Select Text buttons will not be visible since there are no pixels or text labels in drawing windows.

Examples

In the following sequence of illustrations we will make a Select Box mouse selection with various combinations of select objects filter buttons enabled. The Select Box command selects objects that are entirely within the rectangular mouse cursor. We use Replace Selection mode so that every time we make a selection it replaces any previous selection.

The example uses a map with several drawings as layers. It shows land areas in the San Francisco Bay region together with major roads as lines and the centers of named cities as points. The blue water region is not an area but is just the blue background color used for this map that is seen through "empty" space between the land area objects. The task now is to selectively grab just areas, lines or points in various combinations without selecting undesired types of objects.

Each illustration shows the mouse selection box drawn with the Select Box tool overlaid upon the results of the selection to make clear the action of each selection. In real life, the selection would not appear highlighted in red selection color until after we release the mouse button at the end of the selection click and drag.
If none of the three select object filter buttons are pushed in then no objects will be selected.

When all three select object filter buttons are pushed in all areas, lines and points entirely within the cursor box will be selected. Note that only those land areas (representing Angel Island, Treasure Island and Alameda Island) that are entirely within the selection box are selected. The other, larger areas representing the mainland were not selected because they are partially outside the selection box.
If only the **Select Points** button is pushed in only points within the mouse cursor box will be selected. Note that selected points are shown with red color used for their foreground color, the normally black outline of the dots.

If only the **Select Lines** button is pushed in only lines within the mouse cursor box will be selected. Note that only those line objects that are entirely within the selection box are selected. If we wished to select all lines that are touched by the selection box, we could have used **SHIFT - Select Box** for an "open" selection tool.
Poets, artists and those readers with advanced math degrees will have already begun to see a pattern in this example and will have successfully thought ahead to realize that if only the Select Areas button is pushed in then only areas within the mouse cursor box will be selected.

Any combination of the select object filter buttons works to select what we want. If we want just lines and points we push in the Select Lines and Select Points buttons.

A reminder: The illustrations above show both the mouse cursor box as well as the resultant selection at the same time. In real life, the selection appears in red color momentarily after the mouse cursor box is drawn and released.

Select by Type
When a drawing window is active, choosing **Edit - Select by Type** or **CTRL-T** will select all objects of the currently enabled select object modes. This is a quick way to select all lines in a drawing, or all points or all areas. See the **Edit - Select by Type** topic.

### Selections in Drawings Select Records in Tables

If a drawing is linked to a table making any selection in the drawing will also select the records for those objects in the table. When a drawing and a table are linked there is only one selection within either of them because there is a one to one association between records and objects in linked tables and drawings. Selecting an object in a drawing will select (that is, highlight) the associated record in the table and vice versa.

A very important adjunct to making selections in any drawing or drawing layer in a map is using selection within tables at the same time we are looking at the same data in a drawing. It is very important to read the Selection in Tables topic to learn about selection methods in tables.

### Selections Pane

The most frequently used pane when making selections in drawings is the **Selections pane**. We often will use the Selections pane to save selections made in drawings for later use. Each drawing or drawing layer can have up to seven saved selections in the Selections pane. See examples using the Selections pane in the introductory Selection topic.

### Smart Mouse Selection

**Smart Mouse** selection is used to select objects for editing. An object selected with smart mouse selection will appear with edit handles that can be used to move or reshape the object. An object chosen for editing is also called the **primary selected** object.

Commands involving the mouse are normally controlled by whatever toolbar button is engaged. To use **Select Box** for example, we push the **Select Box** button in and then use the mouse in this mode. To provide fast operations when the mouse is not engaged in any other command, Manifold allows "smart mouse" selection in a way that mimics ordinary Windows point and click mouse highlighting and motion methodology as is used in Word and other applications.

### Smart Mouse Touch Selection

The simplest form of smart mouse selection is to click on an object to select it. This is a smart mouse version of **Select Touch**. For example, clicking on an area object in a drawing will select it for editing. Clicking on an object

See the **Editing Objects** topic for examples and more information on interactive editing with smart mouse selection.

### Keyboard Modifiers with Smart Mouse Touch Selection

Because a selection may already exist in the drawing when we wish to select an object for editing, smart mouse selection uses keyboard modifiers to allow a richer set of commands.

**Click** Equivalent to **Select Touch** in **Replace** mode. Click on an object to select it as the primary selected object. All other objects are deselected.

Clicking into an empty part of the drawing deselects all objects.
(with confirmation dialog).

**CTRL**  
Invert the selection state of the object without changing the selection state of any other object. Equivalent to **Invert** mode.

**SHIFT**  
Select the object if it is not yet selected and make it the primary selected object. Does not change the selection state of other objects. Equivalent to **Add** mode.

### Smart Mouse Box Selection

When clicked on an empty region and dragged the mouse automatically makes a box selection analogous to using the **Select Box** selection tool in **Replace Selection** mode. This is a general shortcut for selection. To select an object as the primary selected object for editing, either click on it or **SHIFT** click on it.

### Keyboard Modifiers with Smart Mouse Box Selection

The following keyboard modifiers may be used with smart mouse box selection:

**Click and drag**  
Click on an empty region and drag to make a box selection in **Replace Selection** mode. Selects all objects that are entirely within the selection box.

**CTRL**  
Click on an empty region and drag to make a box selection in **Invert Selection** mode.

**SHIFT**  
Use an open box selection to select all objects any part of which are within the selection box.

### Notes

Why is it called "smart mouse" selection? This is based on the idea that to support this type of selection the mouse cursor in its default mode must understand a lot about its environment. For example, it must be able to keep track of and know when it is clicked on a drawing object (and, for that matter, when the object is in the active layer if the mouse hovers in a map window) and much more. This requires considerably more thought for the cursor than merely being a default Windows cursor, so it is said to be a "smart mouse" cursor.

Note that the default action of the mouse in "smart mouse" action depends on the context of where it is clicked and how it is moved. For example, clicking on an empty region and dragging creates a box selection mouse move. Clicking on an object and dragging is a drag-n-drop operation.

### Tech Tip

When objects in drawings are selected the associated records in the drawing's table will also be selected. One way to browse the data contents of objects is to open both the drawing and the drawing's table.

Click the table window and press the **Filter Selected** button so the table window shows only selected records. We can then select objects in the drawing and see their data contents in the table window.

### See Also

See the Transfer Selection topic to transfer selections between drawings and images.
Linked Drawings

Linked drawings are drawings that are created dynamically from geometry data stored in tables or created by queries. The table or query may be in the Manifold project or it may be a table or query in an external database provider. Linked drawings are shown in the project pane using an icon that include a yellow "database" cylinder to show they are created from a database table or query.

There are two types of drawings in Manifold:

- **Regular drawings** - These are static components in which all geometry (the information that defines the shape, size and location of objects) is stored in the drawing component with data associated with each object saved in the drawing's table component. Such drawings are also called local drawings.

- **Linked drawings** - Linked drawings are created from geometry data stored in database tables or generated from database tables by queries. Such databases and queries can reside within the Manifold project or, as is more frequently the case, they can reside in external database servers, so that even though the linked drawing appears to be a component in the Manifold project the actual data to create that drawing comes from the external database.

Linked drawings have characteristics depending on their source of geometry data:

- **Linked drawings from geometry data** - These are fully editable, read/write drawings containing points, lines or areas that are created dynamically from geometric columns in tables. Linked drawings can be linked from a variety of database technologies using a variety of different methods of storing geometry data. For example, drawings can be linked from geometry data saved in OpenGIS style Geometry(WKB) form, ESRI style Geometry(SHP) form or Manifold Geometry form in a very wide range of database products, such as Microsoft SQL Server, Oracle, IBM DB2, MySQL or even within ordinary Microsoft Access .mdb databases. Manifold also supports Oracle Spatial and Locator native databases saving geometry as SDO_GEOMETRY data. See the Geometry in Tables topic.

- **Linked drawings created from queries** - When linked drawings are created from queries, if the query creates geometry data on the fly then the resultant linked drawing will be read-only regarding spatial data such as the shapes and positions of objects (points, lines, areas).

- **Linked drawings created from geocoded tables** - This is a special case of linked drawing provided for the convenience of users when dealing with simple geocoded data in tables. When a geocoded linked drawing is created from a geocoded table, Manifold will automatically create an intermediate query that takes the simple, non-geometry data from the table's coordinate columns and then on the fly will create geometry data from which a linked drawing may be created. Geocoded linked drawings contain only points or lines that are created from X and Y (Longitude and Latitude) values saved in coordinate columns in database tables, that is, so-called geocoded tables. Linked drawings created from geocoded tables using coordinate columns are read-only and are provided as a convenience to simplify the acquisition and display of geocoded data in a variety of common applications. See the Creating Drawings from Geocoded Tables topic and the Linked Drawings from Geocoded Tables topic for details.

The above guidelines show the tremendous flexibility of linked drawings. However, even with such flexibility we should keep an eye out for logical limitations of various data sources from which drawings may be linked.

For example, linked drawings support adding or removing objects, provided the data source supports adding or removing table records. Suppose we have a table in our project that contains a geometry column. We can link a drawing to that table and it would be possible to add or remove objects in that linked drawing. Suppose now we create a SELECT query in that same project that takes geometry data and other columns from the table. We could also create a linked drawing from that query, but we would not be able to add or remove objects from that linked drawing because it is created from a query, which is a "one way" generator of geometry data.

**Projections**

The projection of a linked drawing is specified by its data source. Attempting to re-project a linked drawing will reload its data from the data source.

**Properties**

The View - Properties dialog for components shows the data source of a linked component (such as a linked drawing) and other relevant information.
The Link / Share dialog accessed from the [...] browse button for linked or shared components accessed from the View - Properties dialog will provide a summary of the link or share properties. This will show the status of a shared component and whether or not changes in a linked component will propagate back to the data source.

**Linked Tables in Linked Drawings**

Like any drawing, a linked drawing also has its own table, which in the case of a linked drawing is a linked table. Such linked tables are just like regular, standalone linked tables and can be used just like standalone linked tables.

For example, we can make changes in the linked table and those changes will be implemented back to the data source from which the table is linked. In particular:

- A linked table allows adding or paste-appending a new column, renaming an existing column, and changing the type of an existing column.
- We can change the design of a linked table.
- It is possible to geocode records in a linked table.

All of the above capabilities apply to both standalone linked tables and tables bound to linked drawings.

Manifold will also try to exploit the capabilities of the remote data source whenever possible in the case of the linked tables associated with linked drawings. For example, if new columns must be created in such tables because objects are being pasted from other drawings Manifold will try to create such new columns as NULL-able (that is, able to contain NULLs).

**Tech Tip**

Importing or linking a drawing from another drawing in the same .map project using the Geom (i) intrinsic field will inherit both the coordinate system as well as the location precision of the source drawing.

**See Also**

- Creating Drawings from Geocoded Tables
- Create a Linked Drawing from a Geocoded Table
- Formatting Lines in a Linked Drawing
- Geometry in Tables
- Importing and Linking Tables
- Linked Drawings from Geocoded Tables
- Multi-User Editing of Linked Drawings
- Project Pane - Open Data Source
- Queries and Geoms
- Spatial DBMS
- Spatial DBMS Facilities
- Spatial Extensions
- View - Refresh Data
Linked Drawings from Geocoded Tables

For a general introduction to geocoded tables, see the Creating Drawings from Geocoded Tables topic.

Manifold can create linked drawings from geometry data in tables or from geometry data fetched or created by queries. Using geometry data in tables and then creating linked drawings from that data allows storing drawings within tables in a fully read/write and editable way. However, at times we deal with simple, non-geometry data in tables and would like a simpler way of displaying such data in drawings.

For example, a **geocoded table** contains two coordinate columns that for each record give the longitude and latitude (or X and Y) position of that record. A classic example might be a table downloaded from a GPS device that provides a list of waypoints with the longitude and latitude of each waypoint. Such a table, because it provides a location for each record in the longitude and latitude columns, is called a **geocoded table**.

For the convenience of users, Manifold provides a simple way of displaying data from geocoded tables within linked drawings. We can create a linked drawing based on a geocoded table, and when we do so Manifold will automatically create an intermediate query that takes the simple, longitude and latitude data from the table's coordinate columns and then on the fly creates geometry data from which a linked drawing is created. Like all linked drawings, the linked drawing is created from geometry data; however, in this case the geometry data from which the linked drawing is created comes from the intermediate query. The automatically-created query takes data from the coordinate columns in the table and creates geometry data necessary for a linked drawing. There is nothing special about this process that could not be done manually, if we were so inclined.

The only magic in this process is that if we try to link a drawing from a table that contains no geometry data (Manifold can see right away if a table contains geometry data or not), then Manifold assumes we want to create a linked drawing from a geocoded table and so it launches the appropriate dialog. When we use that dialog to specify which columns are to be used as coordinate columns (that is, for the longitude and latitude columns or X and Y columns), then Manifold knows what columns to use in the query it writes for us.

If we had good knowledge of spatial SQL we could have written the query manually and then manually linked a drawing from that query. When Manifold does this for us automatically it saves time and requires less thought.

**To create a linked drawing from a geocoded table:**

1. Choose **File - Link - Drawing** and open the desired database file or connect to the desired data source.
2. Choose the table desired and check the desired fields. Specify the fields that contain latitude and longitude values (by default, the system will guess that any fields named "Latitude" and "Longitude" are the fields to use. Press **OK**.

Four new components will appear in the project:

- A linked table - This linked table is a way of linking the table data from the geocoded table that will provide the data for the linked drawing.
- A query - This query transforms the ordinary, geocoded data in the coordinate columns into geometry data from which the linked drawing will be created.
- A linked drawing and the linked drawing's table - The linked drawing is like any drawing, except it is created dynamically from the data within the geocoded table.

For example, if the table below were located in some database provider (such as SQL Server or Oracle) or was saved in a database file such as an .mdb file it could be used to create a linked drawing.

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>1.1</td>
<td>Oil Well</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>Oil Well</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>Oil Well</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>Oil Well</td>
</tr>
<tr>
<td>3</td>
<td>9</td>
<td>Oil Well</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>Gas Well</td>
</tr>
</tbody>
</table>
If we created a linked drawing from the table above using the X field for Longitude / X and the Y field for Latitude / Y we could create a drawing of points like that seen below.

For each record in the table, a point has been created. If we add a new record to the table, delete a record or change an X or Y value then the corresponding point will be added to the drawing, or deleted from the drawing or moved within the drawing.

For a step-by-step example, see the Create a Linked Drawing from a Geocoded Table example topic.

**Uses for Linked Drawings**

Linked drawings allow us to create drawings from data that is changeable and most conveniently stored within geocoded tables or queries. The classic use for a linked drawing is within a Manifold IMS application such as a vehicle tracking web site. Suppose we have a fleet of trucks or ships or other vehicles and we would like to display the location of each vehicle on a web site map. Suppose we can acquire the location of each vehicle through some other application that receives location data from a GPS in each vehicle and places it into a table in our database. Showing the location of each vehicle in Manifold is as easy as creating a linked drawing using that table.

Because most data acquisition software for vehicle tracking applications is already programmed to update a database table, using linked drawings completely eliminates the need to teach the tracking software package how to interact with a GIS system. All the tracking software need do is update a table, as it already knows how to do.

Linked drawings are also very useful in situations where the number of objects to be displayed is changeable. For example, suppose we work for an insurance company that sells its insurance products through independent agents. The list of approved agents might change every few weeks as new agents are added and as poorly performing agents are dropped. We want to create a web site that a prospective customer can use to find the nearest agent offering our insurance products. It would be very convenient to use a linked drawing in the web site so that the list of approved agents can be kept in the form of a geocoded database table.

This reduces the task of updating the list of agents to an interaction with the database table, which allows greater flexibility in software that can be used to update the status of each agent, what products he or she offers from our company and so forth. There will always be many more software packages and programmers who know how to interact with a popular DBMS product than will know how to interact with GIS software. Therefore, using linked drawings is a good way to add dynamic updating of data to Manifold projects and to Manifold IMS web sites without requiring a lot of GIS expertise on the part of accessory software or web site programmers.

**Lines in Linked Drawings**

Although there is no standard GIS method for representing lines in geocoded tables, Manifold uses a simple convention that can be used to create lines in linked drawings from tables. When importing or linking a drawing...
from a geocoded table, we may optionally specify a LineID column that will be used to identify which points will be used to draw lines.

As Manifold scans through the table it will draw lines through all contiguous sequences of two or more records for which the LineID value is the same. When the LineID value changes Manifold will start a new line. If the LineID value is the same for only one record the record will be treated as a point and no line will be drawn. The column used for LineID may be of any type, for example, it could be a numeric or a text column. Newly-created lines will inherit the LineID values of the points used to create them.

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>Comments</th>
<th>Line Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>1.1</td>
<td>line 1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>line 1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>line 2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>9</td>
<td>line 2</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>single point</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>line 3</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>line 3</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>line 4</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>line 4</td>
<td>5</td>
</tr>
<tr>
<td>6.2</td>
<td>5.2</td>
<td>line 4</td>
<td>5</td>
</tr>
<tr>
<td>6.2</td>
<td>5.8</td>
<td>line 4</td>
<td>5</td>
</tr>
</tbody>
</table>

For example, consider the table above. If the Line Number field is specified as the field to use for LineID then Manifold will use the first three records to draw a line through three coordinate points because the value of 1 is the same in the given LineID field. Next, the system will draw a line through two points corresponding to the records with LineID values of 2. The next record has a LineID value of 5 but because the LineID value changes immediately to the next record it is ignored.

The system then encounters two records with LineID of 6 so it creates a third line through two points. Finally, Manifold reads four records with a Line ID of 5 so it creates a fourth line through four coordinate points.

From the above example note there are two key characteristics of this simple format for representing lines in tables:

- Lines are created from records taken in order. The geocoded table must list the coordinate records in the order that the line is to be built, from beginning to end.
- Lines are denoted by a change of value in the LineID field and are not tied to unique values. In the example above the 5 value was used twice in the table. Once it was used for a single record and thus ignored and the second time it was used for a line consisting of a sequence of four points.

To amplify the second point, the table below would have the same effect in creating lines as the table above.

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>Comments</th>
<th>Line Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>1.1</td>
<td>line 1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>line 1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>line 2</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>9</td>
<td>line 2</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>single point</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>line 3</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>line 3</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>line 4</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>line 4</td>
<td>1</td>
</tr>
<tr>
<td>6.2</td>
<td>5.2</td>
<td>line 4</td>
<td>1</td>
</tr>
<tr>
<td>6.2</td>
<td>5.8</td>
<td>line 4</td>
<td>1</td>
</tr>
</tbody>
</table>
A new line would be created whenever the field used as the LineID field changed from a 0 to a 1 or back to a 0. For legibility reasons, though, most applications will use a field for LineID that more naturally identifies the lines to be created than a pattern of changing from 1's to 0's.

For example, if our application is a vessel tracking application that shows where ships are currently located and where they have been an easy way to show the tracks of ships where each ship has a unique ship number is to use the Ship Number as the LineID field.

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>Comments</th>
<th>Ship Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>1.1</td>
<td>line 1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>line 1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>line 1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>line 2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>9</td>
<td>line 2</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>single point</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>line 3</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>line 3</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>line 4</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>line 4</td>
<td>5</td>
</tr>
<tr>
<td>6.2</td>
<td>5.2</td>
<td>line 4</td>
<td>5</td>
</tr>
<tr>
<td>6.2</td>
<td>5.8</td>
<td>line 4</td>
<td>5</td>
</tr>
</tbody>
</table>

We would set up our database so that the table used for the linked drawing is first sorted by Ship Number and then sub-sorted by a date/time field (not shown in the example above) that gives the time each location was acquired. Using the Ship Number field in that case as the LineID field will automatically result in lines that show the ship's track. If we like, we can even format the linked drawing so that lines use directional arrowhead styles so we can see the direction the ship is headed.

For convenience, if we import or link a drawing from a geocoded table and we build lines using a LineID column Manifold will propagate that LineID column to the created line drawing.

**Controls**
When creating a linked drawing the typical dialog seen when connecting to a database file or data source is seen above. The illustration shows a connection to an Access .mdb file data source that was created in the Data Source dialog (as in the example below in this topic). This dialog allows us to specify how the linked drawing should be created from the table.

**Data source**  The source used from the Data Source dialog.

**Type**  The type of geometry source. In the above example it is a Table with coordinate columns, that is, a geocoded table. In more advanced usage we might link a drawing from a table that contained geometry data in various forms, such as with a spatial DBMS or a table containing geom types.

**Source**  The table to use, if there is more than one table in the data source.

- **Select All** - Check all columns.
- **Select None** - Do not check any columns.
- **Select Inverse** - Uncheck all currently checked columns and check all other columns. A fast way to use all but one column: click Select None, check the one column not desired and then click Select Inverse.

(Columns pane) Check columns that are to appear for each point in the linked drawing.

- **X**  Choose a field to use for X coordinate or longitude.
Choose a field to use for Y coordinate or latitude.

**Latitude / longitude coordinates**
If checked, interpret the X / longitude and Y / latitude coordinates as unprojected latitudes and longitudes using decimal degrees. Checked by default. Uncheck if the table coordinates contain projected coordinates.

**LineID**
 Optionally specify a field to use to determine points that are to be used to guide the creation of lines. A line will be drawn through all points having the same value in the LineID field. A new line will be started for each new value in the LineID field. Single-point lines are ignored. The LineID column may be of any type.

**Coordinate order**
 Only enabled if a LineID field is selected. Specify a field to use to determine the order in which records are taken to construct a line.

**Coordinate Order**

When building lines using some LineID column, Manifold by default will take records as they come in whatever order they are presented by the data source for the table. That is risky, because the order in which records are presented might be varied by sorting, compaction or other activities with tables and databases.

Specifying some field that gives the order in which records are taken for lines will guarantee the line is assembled in correct order.

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>Comments</th>
<th>Ship Number</th>
<th>Line Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>1.1</td>
<td>line 1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>line 1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>line 1</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>line 2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>9</td>
<td>line 2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>single point</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>line 3</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>line 3</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>line 4</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>line 4</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>6.2</td>
<td>5.2</td>
<td>line 4</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>6.2</td>
<td>5.8</td>
<td>line 4</td>
<td>5</td>
<td>4</td>
</tr>
</tbody>
</table>

For example, the sample table used above could have a Line Order column added to it that has numbers in ascending order for each of the coordinates of a line. Even if the table is resorted into random order the lines will still be assembled correctly if the Line Order field is specified in the Coordinate order box.

**Refresh Data**

The data in an external table from which a linked drawing is created controls the contents of a linked drawing. When a record is added to or deleted from the table a corresponding point in the linked drawing will be added or deleted. If the coordinates of a record are changed in the table the corresponding point in the linked drawing will move.

When Manifold is used interactively, linked drawings are updated whenever they are opened or whenever we press the View - Refresh Data command. When Manifold is used in Internet Map Server mode, linked drawings will automatically be refreshed on whatever interval has been specified.

Refreshing a linked drawing will fail gracefully in case of a connection error, for example, as might be caused by a network failure if the drawing is linked from a data source accessed over a network. In general, whenever an update fails for a component linked from an external table Manifold will report any detailed error data returned by the data source.
Note that unlike map server mode, in interactive usage linked drawings are not automatically refreshed. They are only refreshed (updated to any changes in the controlling table) when the drawing is opened or when the Refresh Data command is issued. The reason for this behavior is to guard against situations where the linked drawing suddenly changes while we are in the middle of an editing command or a transform operation.

Because external tables can be edited independently of what we are doing in Manifold, if linked drawings were automatically refreshed in interactive mode we could easily encounter a situation where we are doing something and suddenly points might appear or disappear or move about. Suppose we were in the middle of transferring a selection from the linked drawing to an image: in that case the sudden change would wipe out the operation.

Relinking a Drawing Linked from a Geocoded Table

Drawings linked from coordinate columns in geocoded tables cannot be relinked if a connection is lost using the Relink command: attempting to relink such a drawing using the Relink command can only reconnect the drawing to the intermediate query. If the connection to an external data source is lost, the linked table within the project upon which the intermediate query operates is the component that must be relinked. In such cases it is usually easier to simply re-create the linked drawing directly from the external data source using the File - Link - Drawing command.

Cascading Links and Refresh Data

Manifold allows cascading links, where one linked component is created from another linked component. Circular links are resolved automatically. For example, suppose we create a linked table called Table A in a project that is linked from a geocoded table called Table X in an external data source, such as a table in a SQL Server or Oracle database. We could then create a linked drawing in the project that is linked from that linked table. Let's say we call that linked drawing Drawing A.

However, when a cascading link is created in this way, refreshing the linked drawing will update the drawing to any changes made in the linked table, but it will not cause a cascading refresh backwards out of Manifold to the external database. That is, if we make any changes in Table A and we then refresh Drawing A, the drawing will show those changes. However, if we make any changes in the external Table X and then we refresh Drawing A, the changes will not appear in the drawing. We must first refresh Table A so that it incorporates the latest changes in the external Table X and then we can refresh Drawing A.

Properties

We can get useful information on a linked drawing by right clicking on it in the project pane and choosing Properties to see the View - Properties dialog. Clicking the [...] browse button to the right of the Status information line opens a very useful dialog that shows the data source of a linked component, the link technology and the name of the rowset, if the component is linked from a table or query and the behavior of the data in multi-user editing scenarios.

Caching

Linked drawings are automatically cached locally using techniques similar to those employed in Manifold Enterprise Edition to cache shared components that are linked from an Enterprise server. If a connection to an external database table is lost during a work session Manifold will continue to operate the linked drawing using the local cache. When a project containing a linked drawing is saved, the local cache will be saved into that project.map file. When the project is opened again, the cached image will be synchronized with the current version of the database table so when the drawing is opened it will show the latest data in the table.

Use with IMS

Linked drawings work very well with IMS (Internet Map Server) applications. There are three main considerations to keep in mind when using linked drawings within a map server project:

- Choose a suitable refresh interval. The Refresh linked components every ... minutes control in the Export Web Page dialog allows choice of refresh interval. Choosing a value of 0 will force a refresh with every browser session served. With a large number of users, this is a guaranteed way of achieving unacceptably poor performance. A better idea is to choose a low value such as one or two minutes, if a linked drawing must be kept reasonably up to date as, for example, in vehicle tracking applications.


The Internet user login (IUSR_systemname) for the machine running IMS must have access permissions to read the database table used to control the linked drawing. When Manifold runs in IMS mode it runs with the Internet user's permission and so must have the ability to at least read the table.

The path to the table must be preserved, or IMS will not be able to find the data source and open the table.

**Absolute Paths**

When specifying a database table to control a linked drawing the path to the data source is often normalized by Microsoft OLE DB or ODBC drivers to an absolute path. Therefore, when moving projects and any tables used by linked drawings to a different machine, try to preserve the absolute path to the table. For example, if the table is located at C:\Manifold Projects\mytable.mdb on one machine then place it in the same location on the other machine. This is obviously an important consideration when moving files from a development machine to a production IIS machine for use with Manifold IMS in a map server application.

If we would like to exchange data with other users we should employ linked drawings with care if the linked drawing is created from a table or query outside the Manifold project. If the table controlling a linked drawing is outside of Manifold it will not stored within the Manifold.map project file. Sending a colleague a copy of our .map file will not also send them the database table from which the linked drawing is created.

**Re-attaching a Linked Drawing to a Data Source**

If a linked drawing is disconnected from its data source, it may be reconnected using the Relink command. See the Drawing - Relink / Unlink topic.

Linked drawings are deliberately designed to have a very damage-resistant connection to their originating tables. We can even delete the table and the drawing will still function from cache if at all possible, ready to be relinked. Renaming or removing the table or query (or renaming or removing the coordinate columns in the table or query) harms the link between the drawing and the table, but linked drawings are designed to survive such scenarios and can even be relinked to modified data with the Relink command.

While it is not a good idea to tempt fate by deleting tables or otherwise removing data upon which linked drawings depend the robust nature of the connection combined with use of the Relink command will often allow linked drawings to survive significant damage.

**Converting a Linked Drawing to an Ordinary Drawing**

Convert a linked drawing to an ordinary drawing by right clicking on the linked drawing in the project pane and choose Unlink. This will cut links to the controlling table or query and recreate the drawing as an independent, ordinary drawing. See the Drawing - Relink / Unlink topic.

**Importing an Ordinary Drawing from a Geocoded Table**

Drawings may also be created from geocoded tables by using File - Import - Drawing and selecting a database table file or data source in the Files of type box. In this case a drawing of points will be created from the geocoded table but it will not be linked to the table.

**Editing Linked Drawings**

Although a linked drawing that appears in the project pane is normally read-only, it is usually possible to edit that linked drawing by editing the linked table within the project from which it is created, which will end up editing the table in the data source as well. We could also edit the original data source. For example, suppose we have a linked drawing created from a query that grabs data from a table in an external database file. We could change that linked drawing by editing either the external table or by editing the query.

Linked drawings normally are used not for the purpose of interactive editing through the Manifold console graphical user interface but rather to display visually data from some database table. The assumption is that the database table from which the drawing is created will be edited through the database management system or by other programs which will change the data in the table.

**Open Data Source Command**
Right clicking a component that is linked from an external data source in the Project pane and then choosing **Open Data Source** will open the Database Console and connect to the component data source.

Opening a component linked from an external data source in its own window, and then choosing the **Open Data Source** command in the component menu (for example, choosing **Drawing - Open Data Source** when a linked drawing window is open) does the same thing.

**Creating a "Lines-only" Display**

When creating lines in a linked drawing using the **LineID** option Manifold will draw all points as well as lines in the drawing.

This can result in a cluttered display, like that seen above created from the tables used as examples earlier in this topic. The lines have been formatted using a directional line style to show their beginnings and ends. Since points are also drawn they appear above the lines in the same drawing and depending on placement can clutter the drawing.

In interactive work we can easily remove the points, if desired. When working with IMS we cannot interactively delete the points, but we can use a convenient formatting trick to prevent them from appearing. We simply format the drawing so that points are drawn using transparent color for the foreground color.

This causes the entire point to disappear so no points appear in the drawing, resulting in a lines-only drawing. If we need to create an IMS display that has a combination of some lines and some points we can create two linked drawings in our project from the same table, one of which uses the **LineID** option and has foreground formatting for points set to transparent color so there are no points and the second drawing of which does not use the **LineID** option so there are no lines. By using the two drawings in a map with the lines-only drawing appearing in a layer above the other drawing we can cause points to appear below the lines.
The appearance in IMS will be the same except that the points will appear to be drawn below the lines.

There are many possibilities to create exactly the display desired. For example, if a display is desired that shows lines for all points but which draws points only for single points or only for the last point in a line sequence we can use SQL to help create such a display. We begin by creating a lines-only display in one linked drawing. Next, assuming that points have fields that can be used to distinguish one point from another we can use a SQL query in our database to create a table that has only single points or terminal points (through clever use of fields that distinguish one point from another) and then create a linked drawing from the query.

Combining the two drawings in a map will result in a display that shows lines for all points but draws only those points that are single points or that are terminal points. The exact SQL expression that creates a table of only single points or terminal points will, of course, be dependent on what fields are available and how the geocoded table is written in the first place. For example, if we have a vehicle tracking application the code that updates the table with the latest position point may write a flag field that marks this record as the latest in the sequence for that vehicle.

Example

In this example we take a geocoded table of restaurant locations and we create a linked drawing. We then create a query using the table and create a linked drawing from that query.
The table seen above in Microsoft Access shows sushi restaurants in Palo Alto and Menlo Park near the Northern California USGS facility in Menlo Park. The table is similar to the geocoded table created in the Street Address Geocoding topic example.

To create a linked drawing using this table we choose **File - Link - Drawing** and then in the resulting Data Source dialog we must add a new data source for the Access .mdb file in which the table is stored.

Click on the **Add Data Source** button.
In the resulting Connect To dialog we navigate to the folder in which the .mdb file is stored and choose MDB Files in the Files of type box. We can then click on the desired .mdb file to highlight it and then press Open (or, simply double-click on the desired .mdb file).

This adds the file as a data source. We enter a memorable name into the Name field for the data source, in this case, Sushi Restaurants and press OK.
In the resulting Link Drawing dialog we can accept defaults, since the table already has fields named Latitude and Longitude and Manifold will know to use them by default as the Y and X fields. Press OK.

The result is that several new components appear in our drawing, including a new linked table, an intermediate query and a new linked drawing and its table.
If we open the linked table we see it is simply a table linked to the external Access .mdb table. If we change anything in that external table the changes will appear in this linked table and vice versa.

If we open the drawing we see that it contains a point at the location of each restaurant. This drawing has been created from the table via an intermediate query that creates, on the fly, geometry data from the geocoded table.

There are many things we can do with geometry data. However, even though our new linked table (seen above) has been created automatically by Manifold to make it more convenient for us to create linked drawings from external geocoded tables, it is still a Manifold table just like any other. We can use it just as we might any other table.

In fact, we can create queries from that table in addition to the intermediate query already automatically created for us by Manifold. If we would like to customize the data taken from that table we can easily do so with a query.

Suppose that we would like to create a linked drawing that shows only sushi restaurants in Menlo Park. We can do that by writing a query and then creating a linked drawing from that query.

We create a query with the text above. If all we are interested in are the locations of the sushi restaurants we don't need to select everything (using Select *) but we must select at least the Latitude and Longitude fields. In this case, we select everything so that the linked drawing's table will have all the fields that were in the original table.
If we are curious to get a tabular view of the query's results, we can highlight the query in the project pane and press the **Run** button to see the above. Note that records from Menlo Park have been selected.

Manifold allows us to use the results of queries like tables. If we like, since the above results provide geocoded table data, we can create a linked drawing from them. In this case, instead of linking a drawing to an external geocoded table we will link it to a query within the same project.

We once more we launch the **File - Link - Drawing** dialog and choose **This Project** in the **Data Source** dialog. This data source is "built into" the Data Source dialog when linking a drawing. It allows us to create linked drawings from the Manifold project with which we are working.
In the **Source** box in the **Link Drawing** dialog we choose **Menlo Park Restaurants**, the name we used for our query. We accept all other defaults and press **OK**.

Several additional components are created in the project pane: a new linked table called **Menlo Park Restaurants 2** that is linked from the query, an intermediate query, and a linked drawing and its table.

It may seem to be somewhat extravagant of Manifold to create these new components, but doing so provides a very systematic, regular approach to the matter that is the same whether we link a drawing from a geocoded table from outside the project or from inside the project: in both cases, the first thing Manifold does is create a linked table to the data, and then an intermediate query and finally the linked drawing and its table.
When we open the new linked drawing and pan and zoom it to the same location as the other linked drawing, we can see that it shows only those restaurants that are in Menlo Park (which is slightly North and West of Palo Alto).

Now let's do something interesting and show the nature of the linkage between the original linked Sushi Restaurants table and the drawing linked from the Menlo Park Restaurants query. This will show how Manifold can have cascading links within the same project, where one linked component is created from another linked component.

Let's select two of the sushi restaurants in the original table and delete them.

Selecting the two records and pressing Edit - Delete will delete the records for Mikasa and Toshi's Sushiya.
If we click on the Menlo Park Restaurants 2 Points drawing (the linked drawing created from the query) and choose Edit - Refresh Data the linked drawing will refresh back through the query and cause the query to refresh itself from the (now altered) table. The result is that two points, those representing Mikasa and Toshi's restaurants, disappear.

We can bring the records back by clicking onto the original Sushi Restaurants table and choosing Edit - Undo. This restores the records, which has the same effect as if we added two new records.

If we click onto the Menlo Park Restaurants 2 Points drawing and choose Edit - Refresh Data once more we see two new points appear for the restored records. If only it was that easy to bring a good sushi restaurant back into business!
Drawings

The point of the example above is to show how linked drawings are created on the fly from geocoded tables or queries that contain geocoded data. As records are edited, added to or deleted from the original tables or queries, points corresponding to those records will move, appear and disappear as the drawings linked to those tables or queries are refreshed.

Note that when linked drawings are created using cascading links that ultimately arise from data sources outside of the Manifold project, a refresh of the drawing will cause a refresh all the way back to the original, external data source, to update the drawing using the latest data available.

Read / Write Linkage to Databases

Keep in mind that when Manifold links a table from a database that unless something about the connection makes it read-only (not a typical situation), the connection will be read / write. Changes we make in the linked table are changes in the database from which it is linked.

Consider the Sushi Restaurants linked table as it is at the end of the above example.

If we opened the originating .mdb file in Access we could see the same table in Access, as seen above. [We can have the same table open both in Access and also have Manifold running with the linked table opened in Manifold as well.]
Suppose that back in Manifold we highlight two of the records...

<table>
<thead>
<tr>
<th>Latitude</th>
<th>Longitude</th>
<th>Name</th>
<th>Address</th>
<th>City</th>
<th>State</th>
<th>Zip</th>
</tr>
</thead>
<tbody>
<tr>
<td>37.4466</td>
<td>122.1600</td>
<td>Sushi Ya</td>
<td>300 University Ave Palo Alto CA 94301</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>37.4457</td>
<td>122.1633</td>
<td>Miyaki</td>
<td>140 University Ave Palo Alto CA 94301</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>37.4454</td>
<td>122.1610</td>
<td>Higashi West</td>
<td>632 Emerson St Palo Alto CA 94301</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>37.4556</td>
<td>122.1866</td>
<td>Naon Sushi</td>
<td>1328 El Camino Real Menlo Park CA 94025</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>37.4561</td>
<td>122.1738</td>
<td>Tengu Sushi</td>
<td>700 Welch RD Palo Alto CA 94304</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>37.4437</td>
<td>122.1633</td>
<td>Minkichi</td>
<td>550 University Ave Palo Alto CA 94301</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>37.4439</td>
<td>122.1600</td>
<td>Le Poisson Jap</td>
<td>642 Ramona St Palo Alto CA 94301</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>37.4468</td>
<td>122.1625</td>
<td>Jidai-Ya</td>
<td>330 Lytton St Palo Alto CA 94301</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>37.4526</td>
<td>122.1810</td>
<td>Akasaka</td>
<td>925 El Camino Real Menlo Park CA 94025</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Suppose that back in Manifold we issue an **Undo** command to restore the two deleted records to the table. Not only will they re-appear in the linked table in Manifold...
...but will also appear back in the table in Access as well.

Example Scenario

Linked drawings created from tables or queries have great flexibility. In addition to the above example, consider this scenario in which a linked drawing is used to create a particular display:

For example, we might have a table of populated places linked from an SQL Server database. We want to label the places so that labels for larger places appear on top of labels for smaller places.

We create a query for each population range such as...

```
SELECT [Latitude (I)], [Longitude (I)], [Pop]
FROM [Places] WHERE [Pop] BETWEEN 10000 AND 50000;
```

... and we link a drawing from each query.

We then create a label component for each linked drawing and place all the label components in a map so that the label layers for larger places are higher on the display stack than those for smaller places. Each time the project is opened, the system will connect to the table on the SQL Server, re-compute the queries, update the drawings linked to the queries and update the labels bound to the drawings.

Important Note when Using 64-bit Manifold Editions

Due to a lack of required Microsoft facilities in 64-bit Windows systems, Manifold in 64-bit mode cannot export, import, export or link to DB, HTML, MDB, XLS or WKx format files. This includes no access to the MDB parts of Manifold MFD and MapInfo TAB imports. The workaround for importing or exporting such files is to launch Manifold in 32-bit mode by using the Manifold System 7x (32-bit) shortcut, perform the export from or import into a .map project file and then re-launch Manifold in 64-bit mode using the Manifold System 7x (64-bit) shortcut. Linking is more complex: the data must be in some format usable in 64-bit mode within 64-bit Windows systems, such as a SQL Server database. Alternatively, the data can be kept within a Manifold .map project file and linked using the Manifold ODBC driver.

Tech Tip

Master the use of linked drawings with geocoded tables that use straightforward, unprojected longitude and latitude coordinates before attempting to work with geocoded tables that use projected coordinate systems for their X and Y coordinates.

Notes

Refreshing a linked drawing created from a table that resides in an external data source will refresh the table. Refreshing a drawing linked from a query will recompute the query.
Refreshing a query will refresh all linked drawings descended from that query.

Although this documentation is written to require decimal degrees notation, in fact if coordinate columns in a geocoded table use degree - minute - seconds notation Manifold will try to parse the notation used to extract valid longitude and latitude values. manifold.net strongly recommends using decimal notation to avoid any possible ambiguity.

See Also

Linked Drawings
Creating Drawings from Geocoded Tables
Create a Linked Drawing from a Geocoded Table
Formatting Lines in a Linked Drawing
Geometry in Tables
Importing and Linking Tables (for a discussion of ODBC and file types used for tables).
View - Refresh Data
Project Pane - Open Data Source
Multi-User Editing of Linked Drawings

Manifold Enterprise Edition and higher versions support concurrent, multi-user editing of linked drawings that are created from geometry columns in tables. A drawing can be edited by more than one person at the same time by storing that drawing as a table in a database and then working with it in Manifold as a linked drawing. Each user who needs to edit the drawing can link it into his or her project. It may then be edited by multiple users simultaneously, whether the drawing appears in a drawing window or as a layer in a map window.

For example, a drawing may be linked from an Oracle Spatial database into a project on our machine and we could edit that drawing at the same time that a colleague is also editing it on a different machine. If any editing conflicts arise with other users, say if we delete a particular area while our colleague is moving it to a different location, we can resolve such editing conflicts using the Review pane.

Resolution of editing conflicts is important because when potentially many people are editing the same drawing at the same time it is possible that more than one person might edit the same object (that is, the same point, line or area) differently. Manifold uses a version column in the database table to keep track of which objects have been changed.

If we do not have Enterprise Edition or if we do not create and use a version column, we can still use linked drawings in multi-user environments. However, if no version column is in use then any changes made to objects made by any one user will immediately apply to the data source without notice to other users. Other users will only become aware of such conflicting changes when they refresh their linked drawings and see that someone else's edits have been applied instead of theirs.

Manifold Database Administrator Edition provides the Administrator Console to make it easy to prepare drawings stored on database servers for multi-user editing. Administrator Console allows DBMS administrators to designate friendly names for components, to enable storage of formatting with drawings within the database and to specify defaults to be used when importing and linking drawings, such as automatic use of version columns.

Linked Drawings and Refresh Data

Whenever we edit a linked drawing the objects displayed arise from data fetched from the originating data source to which the drawing is linked. The drawing objects we see and edit are cached locally and will not reflect changes in the originating data source until the drawing is refreshed.

Refreshing a drawing using View - Refresh Data or the equivalent keyboard shortcut ALT F5 re-synchronizes the drawing with the originating data source and updates the Review pane with editing conflicts, if any.

Drawings are refreshed automatically when a project is opened (the default setting in the Tools - Options dialog) but otherwise we can work a long time, if desired, on a drawing without doing a refresh. Whenever we do a refresh, our drawing will be synchronized with the data source so that the drawing contains all of the changes made to the data source (by anyone) since the last refresh, and the data source will be updated with our changes. If our changes conflict with any others, we can resolve them using the Review pane.

For this reason, it is wise to Refresh Data or do an ALT F5 when commencing an editing session and then on a regular basis throughout an editing session.

Multi-User Editing Overview

The following guidelines apply to multi-user editing of drawings linked from enterprise-class DBMS products such as Oracle or SQL Server:

- Drawings linked from geometry data support concurrent multi-user editing. Multi-user editing is not supported with linked images, with linked surfaces or with drawings linked from geocoded tables.
- The table from which the drawing is linked must have a writeable version column that is a numeric column. The version column is normally created by the drawing's author when the drawing is exported into the database. Drawings exported to Oracle spatial data will have a version column created automatically.
- We must specify a column to use as the version column version when we link the drawing into our project if we want to be able to cleanly resolve editing conflicts between users. If we do not specify a version column to use, we can still edit the drawing with multiple users at the same time, but Manifold will not be able to resolve multi-user editing conflicts for us. Therefore, it is wise to always use a version column.
• To enable versioned editing (that is, edits using a version column to resolve conflicts) we must include the drawing's ID column (the default setting) when exporting the drawing into the database. Versioned editing uses the ID column as a key column in the database for faster performance. See the discussion below on key columns.

• Manifold provides two modes to use for resolving editing conflicts. The default mode used when a version column has been specified is to interactively review any editing conflicts with changes made by other users. If a version column has not been specified (or, if we simply choose to be authoritative), we can use an edit mode specifying that any changes made by us will overwrite changes made by others without review. The mode in use may be viewed and specified in the View - Properties - Link / Share Status dialog reached from the View - Properties dialog of the drawing.

• We use the Review pane to see the changes we have made as well as the changes made by other users and to choose which version we prefer to use to resolve any conflict.

• Drawings that have had formatting storage enabled on the database by Administrator Console may be linked with formatting from the server. Any formatting changes made by different users will be automatically managed and resolved by the Review pane.

• Most organizations using Manifold for multiuser editing of linked drawings will install some Database Administrator licenses to enable use of Administrator Console to prepare drawings for fast and easy use by all users and to enable formatting storage within database servers.

Version Columns

To make multi-user editing possible we must have a version column in the table from which the drawing is linked. The version column must be a numeric column and it must be writeable. This column may use any name and does not need literally to be called "version," although it is normally it is given a name similar to that. Most organizations using Manifold extensively will also standardize on using a particular name so that everyone knows that a column of that name is intended for use as the Manifold multi-user editing version column. The version column need not be displayed or linked into the drawing's table: when the drawing is linked from the table we do not have to check this column for inclusion in the drawing's attributes.

If we think that a drawing may be edited by multiple users we normally create a version column in the drawing before exporting it into a database system. If a drawing has already been exported into geometry storage in a database table and does not have a version column, we can easily add a version column to that table. To do so, simply link the table into a Manifold project to create a linked table. Open the linked table and right click onto the column heads to add a new numeric column or use the Table - Design command to add a numeric column. Any numeric data type will be fine for the column: most users will employ the 32-bit integer data type used by default to create numeric columns.

Drawings exported into Oracle spatial databases do not need to have a version column created as this will be done automatically during export into Oracle. The name of the version column so created may be customized, if so desired.

Every time an object is changed by someone, the version value for that object in the version column will change. Manifold can compare the version values for any object as stored in the local cache used for editing with the version value stored in the originating table to tell who has changed that object and if any editing conflicts have arisen. Manifold can then apply the edit mode rule specified by the user (by default, to review conflicts) to resolve any conflicts. Conflicts are viewed and resolved using the Review pane.

Specifying Edit Modes

The method used for resolving editing conflicts between users is specified in the View - Properties - Link / Share Status dialog reached from the View - Properties dialog of the drawing. There are two settings:

• Review changes made by others when resolving conflicts. This mode detects changes made by others to drawing objects and allows us to review editing conflicts using the Review pane. We can then decide whether we prefer our changes or changes made by someone else. This is the default mode whenever we use Enterprise edition or above and have specified a version column. This choice enables orderly, concurrent multi-user editing of geometry in drawings.

• Overwrite changes made by others without review. This mode does not detect changes made in drawing objects by others but simply applies any local changes made, overwriting any changes made by others. This is the only choice if we are not using Enterprise edition or if we have not specified a version column. We can still edit geometry in drawings with this choice, but any changes we make will be immediately applied regardless of what someone else may have done to that same object since our last refresh.
The Edit mode applies to geometry editing only: as is the case with any table attribute editing, any edits of non-geometry fields in a linked drawing’s table will always be immediately applied, overwriting any changes made by another user.

**Example**

In this example we will first export the US_Main sample drawing of the United States into an Oracle 10g Express Edition database. We will then simultaneously edit it by two users from two different sessions of Manifold, deliberately introducing an editing conflict. We will then resolve the conflict using the Review pane.

**Step 1: Export a Drawing to Oracle**

We begin by creating a project that has the US_Main example drawing in it, as set forth in many of the topics in the Examples chapter. We will export this drawing to an Oracle Express server, so there is no mystery as to what the drawing is or how it got into Oracle.

**Note:** Depending on the version of US_Main in use the drawing’s table may have more than one field as variable length ASCII type. A limitation in Oracle requires us to change the type of such fields (right click on the column header and choose Change type) to a fixed length field of sufficient length to hold the data so that no more than one field is variable length.

We right-click on the drawing in the project pane and choose Export, specifying Data Sources in the Save as type box to launch the Data Source dialog. In the Data Source dialog we create a data source for the Oracle Express server we choose to use. We then double-click that Oracle Express server data source.
In the resulting Export Drawing dialog we choose defaults as seen above.

Note that the Version column box is already filled in for us since Manifold will automatically create a version column when using an Oracle database. Press OK. Close the project.

**Step 2: Link a Drawing from the Database**

Let us now go to a different machine, launch Manifold and link in the US_Main drawing from the Oracle database.

Since this example will use two different sessions on two different computers to illustrate the resolution of multi-user editing conflicts, we will choose a different color Windows color scheme on each computer. One computer will use the default blue Windows color scheme and the other will use a green color scheme. We will refer to the different computers as the Blue computer and the Green computer.

**On the Blue Computer:**

We launch Manifold with a new project.
We launch the Database Console and connect to the Oracle server using the Data Source dialog. We highlight the US_Main drawing just exported and press the Link button. The Database Console is almost always used to link drawings from big databases because (unlike the File - Link - Drawing dialog) it has good facilities to make it easy to find the one drawing, possibly among very many items in the database, we would like to use.

The Import / Link Options dialog opens up. We will use the entire drawing (and not just some subset area of interest). Linking a drawing from Oracle will automatically choose the VERSION field in Oracle to use as our version field.

If we open the linked drawing and zoom into the Southeastern portion of the US we see it is indeed the US_Main example drawing in play. Note that we have the Review pane opened and undocked so that it is conveniently near the drawing window when we make screenshots.

On the Green Computer:

Let's now move to the green computer and launch another Manifold session. We can repeat the above procedure to link the same drawing from the same Oracle server into this second session.
If we open the drawing on the second server, we can see that it is also the same, US_Main, drawing. We have also opened the Review pane on the green computer.

On the Blue Computer:

Moving to the blue computer, we **CTRL-ALT** click on one of the areas (South Carolina) to select it for editing.

We can then **SHIFT** click and drag one of the editing handles to move the entire state. We will drag South Carolina out into the Atlantic.
We then click on any open space in the drawing to deselect the area from editing.

**On the Green Computer:**

When we switch over to the green computer, the first thing we notice is that South Carolina has not moved. Manifold won’t move objects about in real time as people on different machines edit them, because if Manifold did so it would be impossible to reliably edit drawings in an orderly way. If it did so we could be in the middle of an edit and suddenly the object might be teleported out from under our mouse to a different location.

Instead, Manifold only updates objects when we do a **Refresh Data** command (or **ALT-F5** keyboard shortcut) on the drawing. If we don’t **Refresh Data** on the green computer, the drawing on the green computer will not be refreshed and will not show the changes made on the blue computer. Instead, the drawing will continue to display based upon locally-cached data brought into the green computer from the database to create and display the linked drawing.

In the usual course of events, before we set out to edit an object we would do a quick **ALT-F5** to refresh the drawing to see if anyone already has altered an object of interest before we start editing it. However, because we are trying to simulate a situation where two users on two different machines simultaneously edit the same object in different ways, we will not refresh the drawing. Instead we will proceed to edit it.

Because we have not refreshed, the drawing still looks as though no edits have occurred. The Review pane also reports **No editing conflicts**. We begin by **CTRL-ALT** clicking on South Carolina to select that area for editing.
We then **SHIFT** click and drag the state out to the Atlantic again, but this time to a more northerly position.

The moment we release the **SHIFT** click and drag, South Carolina instantly snaps to the new position, still selected for editing. The moment the area moves, a conflict appears in the Review pane and the pane reports 1 editing conflict.
We can click on the conflict in the Review pane to highlight it. When we do so, Manifold shows us the local edit we have made in blue and the remote edit made by another user in an alarming red color. Colors used for local and remote preview can be set in the Tools - Options dialog; however, the default colors play on the psychology of being easy to remember in that it's natural to assume we are always right and that changes made by other users should be shown in red, "error" color.

Since the same object has been edited in different ways by two different users, we must decide whether to use our local version or to use the remote version. We can choose our local version by pressing the Use Local button.
The conflict disappears from the Review pane and the area we have moved appears without any previews, still selected for editing.

We can click on any open portion of the drawing to deselect the area from editing.

**On the Blue Computer:**

Back on the blue computer we can see there has been no change in the display.
Until we either do a Refresh Data or attempt a different edit, the drawing will not be updated and no editing conflicts will be indicated by the Review pane.

If we choose Refresh Data or do the ALT-F5 shortcut, the drawing will refresh and the area will move to the position given it on the green computer, as indicated by the red arrow. [Manifold does not actually show red arrows when an object moves as part of the refresh. The red arrow was added to the illustration to show how the object moved.]

**Note:** The above example assumes Administrator Console has not been used to prepare drawings for simplified linking by setting default properties using the Database Object Properties dialog. When so prepared, drawings may be linked with a simple click of the Link button in Database Console with no need to specify options such as the version column to use.

**Disconnected Links and Editing Conflicts**

Editing conflicts in a linked drawing are stored locally in the .map project file. If not resolved via the Review pane, they will still be available even if the project is closed and then later opened. This allows deferring resolution of a large number of editing conflicts to a convenient time.

When a project that contains a linked drawing is opened, the linked drawing may or may not be automatically connected to the data source for the drawing. Normally, a linked drawing is automatically connected to its data source when the project is opened as this is the default setting for the Refresh linked components after opening file option in the Tools - Options dialog. However, it is possible that the data source is temporarily unavailable (such as, perhaps, during a network failure) or it is possible that the Refresh linked components after opening file option was turned off for some reason.

If a linked drawing in a project cannot be connected to the data source when the project is opened, any editing conflicts for that linked drawing will be inactive until the drawing is connected to the data source. The Review pane will still report inactive editing conflicts in the total number of conflicts reported in the Review pane's status bar, but inactive conflicts will not appear in the pane's list of editing conflicts. The Review pane will still allow discarding all local changes by using the Use All Remote toolbar button.
Connecting a drawing to the data source by using a **Refresh Data** or **Relink** command will refresh conflict data and will make any editing conflicts active again.

When using a data source to create a linked drawing, Manifold will create a **signature** for that data source which contains a description of the data source's key columns. Manifold can then use the signature to tell if the data source has changed substantially or has been replaced with a similar, but different, data source when a linked drawing is refreshed or relinked. When a linked drawing is refreshed or relinked, Manifold will validate the signature of the data source and will throw away all editing conflicts if the signature has changed, logging a message to that effect in the History pane.

**Usage Scenarios**

Conflicts created by concurrent multi-user editing can become intricate. For example, just because we pause to review an existing conflict in the **Review** pane doesn't mean that other people stop working. While we are considering an editing conflict in the **Review** pane some other person might change the object under review yet again, so that whichever way we decide to resolve the conflict there will still be a conflict with that object that will require a second review.

**Example**

Editing conflicts are detected by storing the current local version of each drawing object and comparing that version to the version stored in the data source every time the object gets changed. Whenever the local version of the changed object is not the same as the remote version, Manifold records an editing conflict.

There are two users, Andy and Bill, who edit the same linked drawing from different machines. The linked drawing contains two areas, Thing1 and Thing2. The version number for each area starts with 10. Andy launches Manifold and opens a `.map` project file containing the linked drawing, which refreshes the data for all linked components in the project. Bill does the same.

Andy changes Thing1, which increments its version number to 11. Bill changes Thing2, which increments its version to 11.

Bill now also changes Thing1. This creates an editing conflict, since the local version of Thing1 on Bill's machine is 10 but the remote version of Thing1 is already 11 because Thing1 has already been changed by Andy. If Bill doesn't have the **Review** pane open, he might not notice the conflict or he might not care, preferring to resolve all editing conflicts later on in his work.

Bill changes Thing2 again, which increments its version to 12. There is no editing conflict, since the local version of Thing2 on Bill's machine was 11 and the remote version of Thing2 was still 11.

**Example**

There can be situations where the object participating in the editing conflict is being changed while the person reviewing changes to the object decides how to resolve the conflict. In this case, committing local changes to the object will fail to resolve the conflict. Instead, Manifold will fetch the latest remote version of the object from the data source for comparison with local edits, to allow a decision on which version to use based on the latest remote and local changes.

There are two users, Andy and Bill, who edit the same linked drawing from different machines. The linked drawing contains two areas, Thing1 and Thing2. The version number for each area starts with 10. Andy launches Manifold and opens a `.map` project file containing the linked drawing, which refreshes the data for all linked components in the project. Bill does the same.

Andy changes both Thing1 and Thing2, which increments their versions to 11. Bill changes both Thing1 and Thing2 as well. This created two editing conflicts.

Bill chooses to use his local version of Thing1 in spite of the changes made by Andy. Bill selects Thing1 and presses the **Use Local** toolbar button. This sends Bill's version of Thing1 to the data source and increments its version to 12.
In the meantime, Andy changes Thing2 one more time, which increments its version to 12 as well. Bill is unaware of that change and reviews the outdated version of Thing2. He chooses to use his local version of Thing2. He selects Thing2 and presses the Use Local toolbar button.

Manifold detects that the version of Thing2, version 11, seen by Bill in the Review pane when he decided to use the local version is not the latest remote version of that object in the data source, version 12. Manifold will not blindly honor the Use Local command issued by Bill without giving Bill a chance to compare his local edits to the latest remote version of Thing2. So instead of applying the Use Local command Manifold will refresh the Review pane on Bill's machine with the latest remote version of the object and will keep the editing conflict open. Bill now has a choice between his local edits and the latest remote version of the object.

As seen in the above example, the Use Local command is not a blind, "crush the remote version no matter what" command. It is a more nuanced command meaning "use my local version if what I see in the Review pane accurately reflects the latest status of the remote version at the time I issue the Use Local command and if not, get the latest remote version and show me the conflict again so I can make up my mind."

Multiuser Editing and Formatting

If we have a Manifold Database Administrator Edition license we can use Administrator Console to enable storage of formatting for drawings in databases. Database Administrator Edition must first be used to enable storage of formatting for a drawing saved in the database. Once storage of formatting is enabled then other Manifold instances, for example, an Enterprise Edition license used by some other user can take advantage of formatting storage in the data source for that drawing. However, if Database Administrator Edition has not been used to configure storage of formatting for a particular drawing in the database, then if we link that drawing into a Manifold project we will not be able to change formatting from the default format settings.

When a drawing saved in a database has had formatting storage for it enabled in the data source, more than one user at a time can link that drawing into a project and change that drawing's formatting. Concurrent multi-user editing of linked drawings that involves changes to formatting by different users is similar to how concurrent, multi-user object editing works:

- Refreshing a linked drawing fetches the latest formatting for that drawing stored in the data source.
- Changing the formatting of a linked drawing uploads those formatting changes back to the data source.
- Editing conflicts when two different users change the formatting in different ways is resolved using the Review pane.
- If the data source becomes unavailable in the process of uploading updated formatting data, the changes are saved in the local component and appear as an editing conflict.

See the View - Panes - Review topic for a visual example of multiuser editing with formatting changes.

Supported Database Systems

Manifold's support for concurrent multi-user editing works with drawings linked from any database in which geometry columns can be stored, so long as that database allows multi-user access and the version column used to keep track of changes in the multiuser environment is writeable. Obviously, if the drawing is being edited the geometry column and any attribute columns involved in the editing must also be writeable.

Versioned editing uses the ID column as a key column in the database for faster performance. Therefore, the database must be able to handle key columns. All modern enterprise-class DBMS products and even most consumer-class DBMS drivers likely to be used with Manifold have this capability.

Although Manifold supports concurrent multi-user editing of drawings linked from a wide range of DBMS products, there is no reason not to use an enterprise-quality DBMS like IBM DB2, Oracle or SQL Server. The Express versions of these database products are free and are vastly superior to consumer-grade free data sources such as Access .mdb. Complete, licensed installations of these three DBMS products may be downloaded for free.

This example happens to use Oracle, but all three of the major DBMS products are superb, well-engineered software and will work well with Manifold. All three have a host of features with many sophisticated and subtle distinctions in their approach to enterprise DBMS with extensive documentation and far-reaching user communities in their support.
Users tend to have very strong opinions on which DBMS they prefer, opinions so strong that at times they make political or religious controversy seem pallid in comparison. Therefore, other than choosing the "Big 3" as examples for technical support, Manifold does not take sides over which DBMS is the "best." The best one is the one you decide is right for your needs.

If you would like advice in choosing which DBMS to use for your work with Manifold, consider asking the advice of your colleagues in the Manifold Online Community.

**Limitations with IBM DB2**

IBM DB2 Express-C Edition may be used for Enterprise Edition Enterprise Servers without any technical limitations. However, two small usage limitations apply when DB2 Express-C is used for storing geometry in tables in the database to support subsequent concurrent, multiuser editing. When a drawing is linked from a DB2 Express-C data source the following limitations apply:

- Adding a new object and immediately editing it without refreshing the drawing creates an editing conflict.
- Adding a new object and immediately deleting it without refreshing the drawing will fail.

Therefore, whenever editing drawings linked from a DB2 Express-C data source make sure to refresh the drawing after adding a new object before attempting to edit that object or to delete that object.

**Key Columns**

Whenever we use a version column to help resolve editing conflicts Manifold must rapidly identify any changes caused by different users within the geometry data in the originating table. That requires high speed access using key relationships supported by the DBMS in which the table is stored. If a data source does not have a key, which can be one column or a combination of columns, versioned editing will not work.

The ID column in any Manifold drawing's table is guaranteed to be unique, since the ID field provides a unique numeric identifier for each object. Exporting drawings to data sources such as Oracle therefore automatically designates the ID column as the key column. This works automatically provided the data source supports key columns and the ID column is included in the list of columns to export.

Virtually all enterprise class DBMS products like IBM DB2, Oracle and SQL Server support key columns, as do many consumer class DBMS products. However, certain database drivers, like those for CSV and DBF files do not support key columns. These formats therefore do not support versioned editing within Manifold System.

Although the ID column will be automatically designated by Manifold on export of a drawing to a data source, if we do not export the ID column or if geometry has been sent to a table (perhaps programmatically or as the result of some manual process other than export) we will have to designate a key column to enable versioned editing. Different DBMS products have different ways of specifying a key column.

If we don't have a key column in such data sources we can still edit the drawing if we do not designate a version column when linking the drawing. In that case the drawing will be fully editable but we will lose the protection of editing conflict resolution given by the version column.

However, failing to have a key field in a data source but designating a version column can have some odd consequences. For example, suppose we link a drawing from an Access .mdb file that stores geometry, that had a version column designated when linking the drawing but which does not have a primary key column in the originating table. In that case we will be able to select items and we will be able to edit attribute values but we will not be able to select an object for editing in the drawing.

The solution to avoiding such oddball situations is to always use a DBMS that supports key columns and to always designate a version column. That's easy enough to do whether we are using consumer grade data sources such as Access .mdb files or have everything done automatically for us with an enterprise-class DBMS like Oracle.

To create a primary key in an .mdb table, link the table into Manifold as a linked table and open the linked table. Choose Table - Design to see and alter the design of the table. Add a new field to be the key field using a default 32-bit integer type. In the Table Design dialog toolbar press in the View Extended Properties mode button. In the Unique column that thus appears for the key field, choose primary key as the value. Press OK.
Tech Tips

Remember, ADO.NET connections are read-only. If we are link a drawing from a DBMS table using ADO.NET, we will not be able to edit the drawing. To link a drawing so it is editable we must use some read/write connection technology such as OLE DB or ODBC or, in the case of Oracle, the automatic use of Oracle's OCI.

Projects may contain many linked drawings, which could be linked from different data sources. The Review pane will show conflicts for whatever drawing window or drawing layer has the focus.

FAQ: Why not Auto-refresh Drawings?

Novice users sometimes wonder why drawings are not refreshed in real time in Manifold, with the drawing objects they contain being changed instantly whenever some other user changes them. That's not done with interactive Manifold sessions to preserve the ability of users to organize their work using conceptually unambiguous beginning and end points to tasks, and to preserve the integrity of data against bizarre modifications caused by the unpredictable overlap of complex editing operations launched by different users at the same time.

Keep in mind that Manifold has profoundly greater editing capabilities than simply moving a dot on a map from one location to another, or simply changing the shape of a line or an area. Manifold operations include commands of awesome complexity and power that can change large numbers of objects at once, dramatically alter the topology of objects or their metrics, create or eliminate objects and so on. Consider, for example, what could happen in a sophisticated Topology Overlay command or a Clip with (Intersect) transform if part-way through the operation a different user changed something about the objects that were being used to "cookie cut" through large numbers of other objects in a different drawing layer. The result could be chaos.

It's true that Manifold could provide extensions to the user interface that equipped such commands with accessory dialogs to deal with the many varieties of chaos that might be induced by such interleaved, real-time edits. But that would greatly complicate the user interface that users would have to learn with a host of special cases and complex dialogs. Instead, the current system provides a simple, clear firewall between changes we make and changes others might make.

In all cases with Manifold, "what you see is what you get." What users see in their linked drawing windows is exactly the local data upon which commands operate. To synchronize our local project with the data source from which the drawing is linked we do a Refresh Data or an ALT F5. Each such refresh brings us up to date and also brings the data source up to date with our changes. If any changes we have made since the last such refresh conflict with edits done by other users, we can see such conflicts in the Review pane and in an orderly way deal with them.

However, in all such cases because the various edits have each in their own projects been "firewalled," so to speak, between refreshes done by their users, each such editing change stands as an organic, coherent whole. When we resolve editing conflicts in the Review pane we can have confidence that whether we pick our local edits or those made by someone else they will be applied to the project in an orderly way, without the chaos of partial combinations between something we did and something someone else did.

Note that in Internet Map Server web sites, we can force a refresh of all linked components on a desired time interval. Such applications are either view-only or permit editing only under the fully-custom control of the application developer. That's a very different situation than the sweeping editing commands available as part of the Manifold user interface.

Advanced users might note there is a philosophical conflict between the rationale described above for avoiding auto-refresh of drawing objects and the immediate propagation back to the data source of changes made in ordinary data attributes in linked tables. That's certainly a breach of perfect philosophical orthogonality, but one rooted in long tradition in multi-user databases. DBMS operators and software vendors have long ago evolved mechanisms for dealing effectively with multi-user attribute editing, which still will be in play in the host DBMS whether an attribute is changed by a Manifold session or some other client of the DBMS server.

It's true that in a GIS environment, where attributes can affect the operation of spatial commands such as the Dissolve command, allowing instant propagation of attribute changes does indeed raise the risk of chaotic interleaved edits of drawing objects. However, even in this case the use of the Review pane will alert users to such conflicts because the object modifications caused by such attribute-dependent commands will result in different objects that will be identified as editing conflicts in object geometry.

See Also
Database Administrator Edition
Enterprise Edition
Linked Drawings
Tools - Administrator Console
Tools - Database Console
View - Panes - Review
View - Properties - Link / Share Status
View - Refresh Data
Using Administrator Console
Hyperlinks

Hyperlinks are URLs (Internet hyperlinks) that are embedded into objects in a drawing. If an object has a hyperlink, double clicking on the object will launch an Internet browser using the object’s URL. If an object does not have a hyperlink, double clicking on it will launch the Fields dialog.

Adding hyperlinks to objects:

1. Open the drawing's table.
2. Create a new field called URL that is of type URL or any text type (fixed or variable length ANSI or Unicode text).
3. For each object that is to have a hyperlink, enter the desired hyperlink into the object’s record in the URL column.

When the mouse is double-clicked on an object in a drawing, Manifold goes to the URL field of the table, fetches the hyperlink stored there for that object and launches an Internet browser using that link. If the URL field is empty, or if it is not of database type “URL,” Manifold will launch the Fields dialog for the object.

This provides a fast meaning for double clicking on an object in most cases. If the object has a URL an Internet browser is launched. If it does not have a URL, the Fields dialog pops open.

Example

We would like to use the Congress.map example provided on the Manifold CD to look up the web pages of each US congressional representative by double clicking on their district in a map.

Open the Congress Table associated with the Congress drawing. The table has a column called Web page that contains the URL of each congressional representative’s web page. This Web page field is an ANSI Text field.

First we will rename the field to URL. Right-click on the Web page column head and choose Rename. Change the name to URL.
Next, we will change the field type to **URL**. Right click on the column head and choose **Change Type**. In the **Column Type** dialog, change the field type to **URL**. Press **OK**. Close the table window.

We can now open the **Congress** drawing window. Whenever we double-click on a congressional district in the drawing, Manifold will launch a browser window loaded with that representative’s web page taken from the **URL** field. For example, we can double-click on Nevada (which has only one congressman for the entire state outside of Las Vegas).

Like magic, a browser window pops open with Congressman Jim Gibbon’s web page in it.

If we wanted to see the fields associated with that district, we could have **CTRL**-double-clicked onto Nevada.
That will open the **Object Fields** dialog for that object.

**More sophisticated uses**

The example above shows URLs in use to open a very simple web page. We can also use URLs that are local references to files on our local hard disk. If such files are image files in a format that can be displayed in browsers (such as `.jpg`, `.png` or `.gif`) we can URLs to show pictures associated with objects.

Because web pages can be very rich and browsers can display all sorts of things automatically, we can use the general mechanism of specifying a hyperlink via a URL to display all sorts of things. For example, we could have a drawing that shows a factory floor plan with locations of different machines. For each machine we could have a URL pointing to a PDF document or other document that provides its user manual. Double clicking on the machine in the drawing will open its manual.

More dynamic information can be achieved by linking a table into the Manifold project from an external database and then forming a relation between the linked table and our drawing's table. If the linked table contains a URL field called "URL" and that column is brought in via a relation it will be used for the hyperlink. We could therefore keep a database of sites, documents, images or other information that dynamically changes so that whenever we double-click onto an object in the drawing the URL field to which the browser jumps will be taken from the external database provider's table.

**Tech Tips**

Since most drawings in GIS aren't used for Internet hyperlinks the fallback behavior of hyperlinks to launch the **Fields** dialog is a useful way of preserving double-click functionality for a quick look at object fields. However, in the case of drawings that are used for Internet hyperlinks if a particular object has an empty URL field the **Fields** dialog will also be launched. That may not be desired. To avoid this situation, create a simple web page that reports something like "This item does not have a web site" and save it on the local disk. In the table window select all records with an empty URL field and fill it with the local path to that web site.

For example, if we created an empty.html web page in the C:\tmp folder we would fill all empty URL fields in the table with C:\tmp\empty.html. When someone clicked on an object that did not have a web site the browser would launch using our empty.html web page.

Whether double-clicking an object launches a new browser window or appears within a previous browser window is set by your browser options. In Microsoft Internet Explorer, see **Tools - Internet Options - Advanced** and uncheck the **Reuse windows for launching shortcuts** box if you would like each URL to launch a new browser window.

When using drawings that contain hyperlinks within Manifold IMS applications we will probably also want to use **layer restrictions**
Formatting

Formatting Drawings

Drawings are made up of points, lines and areas that are drawn in a "connect the dots" fashion between exact coordinates. The appearance of drawings is highly dependent on the formatting choices used to draw the points, lines and areas. Drawings may be formatted to provide a particular appearance, or a theme can be created for a drawing that shows that drawing using desired formatting.

The main formatting choices are:

- **Foreground Color**: The color of solid points and lines and the foreground color of two-color area styles or of two-color area border styles. Choosing a transparent color for foreground color for an area will turn off display of that area. This is a short-cut way of turning off areas when desired.

- **Background Color**: The fill color in most styles used for points, lines, areas and area borders.

- **Style**: The pattern or icon used to draw points, lines, areas or area borders.

- **Size**: The relative size to use for patterns used to draw points, lines, areas and area borders. Does not apply to bitmapped images used for points.

- **Rotation Angle**: Allows rotating the symbols used for points to a given angle.

The Format toolbar for drawings shows the basic formatting choices in use for the active drawing or drawing layer in a map.

![Format Toolbar](image)

It shows **foreground** color, **background** color, **style** and **size** currently in use for areas, area borders, lines and points as well as the **point rotation angle** button for points. Default formatting uses light gray for background colors with black used for foreground color for lines and points and dark gray used as the foreground color for areas. Changes made in this toolbar will immediately be applied to the drawing. Drawings (and thus drawing layers in maps) have one set of formats for all areas, lines and points in that drawing. The format toolbar always shows the formats in use for the active layer or drawing.

**To Change Formatting**

1. Click on the layer tab that contains the objects.
2. Click on the format control to be changed (colors, style or size) for the type of object to be changed (areas, lines or points).

**Color Wells**

Click on the foreground or background color wells in the format toolbar to change color.
The color wells display knows what the current color is and will display a sample of color wells that Manifold reckons will be helpful. The screen shot above shows that the current color is a bright blue (indented well). The top line will always show lightness variations of the current color. The second line in the color wells display will show grayscale variations. The X color well in the upper right is **transparent color**. Using that color will result in transparency everywhere the color is used.

The rest of the color wells pane shows a range of colors beginning with blues and greens and ending with reds. Scroll down using the down arrow at the side of the pane.

Press **Theme** to invoke thematic formatting. Press **More** to open the standard Windows color-picking dialog.

**Example**

This example shows changes to area formats, line formats and point formats. Area border formatting will not be changed in this example.

We begin with a map of California containing four layers:

- **CA areas** contains areas showing the land regions of California
- **CA hydro** contains lines showing various hydrological (water) features.
- **CA roads** contains lines showing main roads.
- **US towns** contains points showing the center points of populated places with populations above 20000.
In this example map, all of the layers contain one type of object (areas, lines or points) only. In the illustration above, the CA areas layer is the active layer so the format toolbar shows the formats for that layer (all default settings). Even though the CA areas layer contains areas only, the lines and points format controls are enabled so we can set values that will be used for any lines or points that might be created in this layer. The illustration is zoomed into the southern part of San Francisco Bay, into the heartland of Silicon Valley.

To change the color of the areas in the CA areas layer, we click into the background color well for areas and change the color. Let’s change it to a khaki green color. All areas in the layer will instantly change background color. Note that the default area pattern draws the edges of areas with a pixel of foreground color, which remains the default dark gray.

Let’s make the water features blue. These are located in the CA hydro layer so we click the CA hydro tab to make this layer the active layer. As we click the CA hydro tab, the format toolbar will switch to show the settings for this layer. We haven’t changed anything in this layer so it still shows the default settings. For example, note
that the area background color shows the default gray color since in this layer (unlike the CA areas layer used in the previous illustration) we have changed the area background color.

Solid lines are drawn in foreground color, so we click into the foreground color well for lines and change the color to a bright blue. Instantly, all lines in this layer will change to blue color. Note that this change affects lines only in this layer. Road lines, for example, are not changed when we change the formatting of the CA hydro layer.

To change the format of road lines, we click the CA roads tab. The format bar changes to show the formats for this layer, which are still the default settings.
Let's make the road lines a dark green. We do this by clicking on the foreground color well for lines and changing the color to dark green.

If we want the road lines to be less visually apparent, we can change them to a dashed line style by clicking into the style well for lines and choosing a dashed pattern. The dashed pattern we chose uses foreground color for the dark bits and background color for the light bits. If we choose the same color as was used for the areas in CA areas for the background color for our lines, they will appear to be dashed segments.
A better way of achieving this same effect is choosing transparent color for the background color; that is, to use no color (for complete transparency) for the background color of lines. Click on the transparent color well at the upper right of the main palette to chose transparent color.

If we chose transparent color for the background color, the alternate segments of the dashed line style would be transparent. This would avoid the unpleasant effect seen above where the road lines cross San Francisco Bay but still have the light green background color in alternate dashes. Had we used transparent color the white color of the bay's background would have been visible through alternate dashes and thus the dashed line effect we seek would be preserved no matter what color background.

We now click the **US towns** layer tab to set it active. This layer still uses default formatting as the format toolbar shows.
We can make the points in this layer instantly more visible by changing the background color for points (the fill color) to bright yellow.

Clicking into the style box for points and choosing a square changes the points to square boxes. Increasing the size parameter to 10 makes the boxes larger.
Changing line or area format settings for the **US towns** layer will have no visual effect, since there are no objects in this layer except points. However, if we would like to set formatting to be used for any area or line objects that might be created or moved into this layer we can do so. In the illustration above, we have changed the area colors to a black foreground color and a brick-red background fill color.

If we now use the *Shapes* toolbar to draw an area in the **US towns** layer it will be drawn using the new format colors specified for areas in this layer. In the illustration above, we have drawn a circular area centered on the Gordon Biersch brewery restaurant in Palo Alto. This is a somewhat contrived example, since it is poor organization to mix different conceptual things (center points of towns as well as areas showing different types of zones of interest) in the same layer.
Finally, if we click back on the CA hydro layer, note that none of the formatting changes made to other layers has applied to this layer. It still retains default formatting for areas and points, with the only change being the use of bright blue color as the foreground color for lines. Changes made with the format toolbar apply only to the active layer.

**Points and Lines in the Same Layer**

Drawings are often imported from GIS formats intended for use in maps where the roads might be used for transportation network analysis. In such cases the map of roads is drawn as a network with point objects located at the ends of each line. This is done because mathematically a network is defined by points (nodes) and not the lines (links) between them.

When imported into Manifold, such road drawings will appear in default formatting. Because default formatting uses a size of 3 for points, the roads will have lots of "bubbles" throughout. Changing this is easy.

Simply click on the size well for points and change the size to 1. A size of one draws each point symbol as a single printer's point in size, so the points will blend into the default size of lines.
The above is a quick means of dealing with the visual appearance of the drawing, but it still leaves the points in the same layer as the lines. If we change line colors or size, we will have to remember to change those parameters for the points as well if we want the points to continue to appear blended in with the lines. A better idea is to select all the points in this layer and to move them to their own layer. We can then turn that layer off and on at will whenever we want to see the network node points. Should we want to do transportation network analysis, we can select the lines and points to be used as a network. Manifold has no problem working with nodes and links that are in separate layers.

**Thematic Formatting**

Thematic formatting is the use of a controlling data field to automatically vary the color, style or size of objects. For example, we might vary the size of points showing cities based on a population field for each point. Cities with greater populations would be shown as larger points.

See the Thematic Formatting topic for more information.

See the Color topic for fast "four-color" coloring of drawings.

**Themes and Drawings**

-Thematic formatting is the use of a controlling data field to automatically vary the color, style or size of objects. For example, we might vary the size of points showing cities based on a population field for each point. Cities with greater populations would be shown as larger points.

See the Thematic Formatting topic for more information.

See the Color topic for fast "four-color" coloring of drawings.

**Notes**

- The most common formatting error when formatting layers in maps is to lose track of which layer contains which objects. The best way to avoid this is to name layers in a sensible way ("Roads," "Streams" and so on) so that it's obvious which objects are in which layers. If we lose track anyway, we can always see which objects are in a particular layer by double clicking the layer tab to turn it OFF and ON. It will be immediately obvious which objects are in that layer as they turn OFF and ON.

- If one wishes to uses hundreds of formats in a map to show the same drawing in hundreds of different ways, doing so is perfectly possible. To do so, we create as many themes for the subject drawing as desired.

- To have a different visual appearance for objects in the same layer, use Thematic formatting. This assures there is some guiding pattern to the different colors or other format attributes.

- The hydrography lines in the California map example do not line up perfectly with the edges of areas because they were taken from two different data sets. The hydrography lines are from the 1:2M-scale USGS DLG map series while the CA areas layer was created using NOAA 1:100K-scale shorelines data.

- See www.gordonbiersch.com for more on where to find real beer when visiting your venture capitalists in Silicon Valley.

**Setting Background Color**

Drawings have an overall white background color by default. Background color for all components is set by default in the Tools - Options dialog. Each individual component can have its background color set by opening the component in a window and choosing View - Properties.

**See Also**

- Themes
- Format Toolbar
- View - Properties - Zooms
Transparent Color

Areas, lines and points in drawings are colored using two colors: a foreground color and a background color as described in Formatting Drawings. In addition to use of ordinary colors, a transparent color may be selected as a foreground or background color. The transparent color appears as the upper right choice in the pull-down color palette for color wells in the formatting toolbar. A choice of transparent color is a choice of 100% transparency (zero opacity) for those parts of objects rendered in transparent color.

To choose transparent color, click into the foreground or background color well desired in the format toolbar and then click on the transparent color well at the end of the main palette.

For example, suppose that in a map with two drawing layers we have a circular area in the upper drawing.

If the area is formatted as seen above, the white color used for background color is opaque. The white color prevents objects in the lower drawing from being seen through the "white space" parts of the crosshatch style pattern used for the area.

If we change the background color to transparent color then the transparent color icon appears in the background color well and the background portions of the area style pattern are made transparent. This allows the lower drawing to be seen through those lower portions.
Transparent color may also be used with line and point styles.

**Transparent Foreground Color**

Choosing a transparent color for foreground color for areas, lines or points will turn off display of areas, lines or points respectively. This is a short-cut way of turning off a class of objects (areas, lines or points) in certain effects. This allows a thematic format to be arranged so that foreground color is transparent with certain values, thus turning off for display any objects with transparent foreground color. For example, if a thematic format is defined that uses the *Selection (I)* intrinsic field to show foreground color as transparent if an object is not selected and black if an object is selected, then only selected objects will appear.
**Area Formatting**

Areas may be formatted to set foreground color, background color, style and size for both the interior of the area as well as the area border, a total of eight formatting characteristics that we may specify to control the appearance of areas. Any of these eight formatting characteristics may be automatically controlled via thematic formatting using a data field.

The wells in the leftmost partition of the format toolbar control the formatting of areas and area borders. The formatting controls for area borders are similar to those for the formatting of lines.

**Examples**

It is easiest to understand area formatting by creating an area and then changing formatting settings in the format toolbar.

The illustration above shows an area together with part of the undocked format toolbar. Default colors are used for the area.

We can click into the area border foreground color well and change it to blue. That will cause the area border to appear in blue color and be more obvious.

The default area border size is very thin, only 1/20th of a printer's point. The border will appear thicker, about 1 printer's point, in most computer displays because computer monitors do not usually have the resolution to display less than about one point. However, when printed out in a high resolution printer the area borders may appear to be very thin if the size of the area border is not increased.
We can make the area border thicker by clicking into the area border Size well and changing the size to 5. Note that the area border is not a line object surrounding an area object. It is all the same area object, just with an option to draw the edge of the area in a different manner.

We can change the Style used for the area border to an alternating foreground color and background color dashed line.

We can change the area border background color to yellow to show how the background color is used with the foreground color in this particular area style.

Some area border styles are asymmetric, in that they appear oriented towards the interior of the area or towards the exterior of the area.
We can choose one of the asymmetric area border styles to see the effect.

In this case, the style results in an area border that appears oriented towards the interior of the area.

For a different effect we can choose an asymmetric area border style with the opposite orientation.

This results in an area border style that appears to be oriented toward the exterior of the area.
Let's go back to a plain alternating color area border style. We will click into the area style and choose a dashed horizontal line pattern to get the effect shown above.

Changing the area foreground color to red shows how this style uses foreground color for the horizontal lines.

Changing the area background color to light blue shows the use of area background color.
If desired, we can use these eight formatting controls (foreground color, background color, style and size for areas and area borders) to create strikingly ugly and confusing graphics.

It is best to use simple and understandable formatting for drawings. The area border formatting controls should be used with restraint.

**Technical Note**

Although asymmetric area border styles may appear to be oriented towards the interior or towards the exterior of areas, in actual fact they are oriented to the left or the right of the path of sequential coordinates that define the shape of the area. In the above case the area shown happens to be drawn using a sequence of coordinates proceeding in a clockwise manner so the asymmetric border styles ended up having the orientations shown. The very same area could have been drawn using a sequence of coordinates proceeding in a counter-clockwise manner and in that case the area border styles shown would have had opposite orientations. When the same asymmetric border style is used for all areas in a drawing it is important to have those areas drawn in the same clockwise or counter-clockwise manner for the resulting orientation to be the same in all areas.

**See Also**

* Area Styles and Size
* Areas and Boundary Lines
* Thematic Formatting
Area Styles and Size

When using Windows 2000, XP, or 2003 area styles are resizable. Changes in the size parameter will alter the spacing of the lines used in these patterns. The size parameter is in number of printer's points, the same unit of measure used for font sizes. One printer's point is 1/72 inch.

If sizes are set very small (2, for example), the resultant styles will be drawn with lines so close that they appear to be solid areas. To avoid coating printed pieces with solid dark ink, area size of 1 will default to solid background color.

The following sequence of examples reformats a province in Mexico using size parameters of 1, 2, 4, 8, 16, and 32 with the crosshatch area style. The province being changed is in a map layer above the other provinces so it can be formatted differently. In all cases the foreground color is black and the background color is white. The other provinces in the map have been formatted in a solid style using black as the foreground color. Boundary lines are in green and the map background is blue. Note: A size of 8 is the default.
To change the size used in an area, click on the size box for areas in the Format toolbar.

**Transparent Foreground Color in Areas**

Choosing a transparent color for foreground color for areas will turn off display of areas. This is a short-cut way of turning off areas in certain effects. This allows a thematic format to be arranged so that area foreground color is transparent with certain values, thus turning off for display any areas with transparent foreground color. For example, if a thematic format is defined that uses the `Selection (I)` intrinsic field to show foreground color as transparent if an area is not selected and black if an area is selected, then only selected areas will appear.

**Notes**

Area styles are not resizable in '98 or Millennium. Area styles are fixed at size 5 in those systems since all they support is an 8 x 8 pixel bitmap matrix.

Fractional area sizes, such as 1/20th of a point, are available in the size selection pull-down menu. Such very small sizes might be useful when printing to high-resolution printers, such as 2880 DPI printers, where such tiny patterns might not blend together into solid color.

The `/list` command line option provides a handy way of getting a list of all available formatting styles and their names for use by programmers. See the Command Line Options topic.
Areas and Boundary Lines

Areas are "solid" objects consisting of a defined region. The default area style as well as the area styles most frequently used to format drawings will use contrasting color for the pixels at the very edge of the area. This makes it possible to see different areas when the same color scheme is used for all areas.

In the illustrations above, a "seamless" area style (all solid background color) was used for the illustration on the left. The default style was used for the image on the right. It's clear how coloring the pixels on the very edge of an area with a contrasting color will show the "seams" between area objects that fit together perfectly.

The risk from this visual convenience is that area styles using contrasting color for the edges of areas might make areas seem to be "two part" objects that consist of an interior area object plus a differently colored boundary line. Because area borders, that is the boundary line of an area, may be separately formatted from the main "body" of the area it is especially easy for new users to make the mistake of thinking of an area border as though it is somehow a separate object.

This will especially be the case when transparent area styles are used (as above) or when the background ("fill") colors of areas are thematically formatted with different colors while the foreground color (used at the edge of the area) is the same:

If an area style is used that does not use contrasting foreground color for the edges of areas, we can see that the above illustration is made of area objects only without any lines in use:
In general, when areas in the same layer are thematically formatted so their colors are different it is easy to tell them apart. When they use the same color, it is difficult to tell them apart if a contrasting foreground color for their very edges is not used.

There are times when we do not wish areas to be drawn with an apparent boundary line. Using styles that show areas with an apparent boundary can be confusing when doing topological work involving the creation of areas from enclosing boundary lines or the creation of boundary lines from areas. In such work, it is often helpful to use formatting styles that cannot be misinterpreted as showing objects that do not exist.

Another example of when we might not want to use styles showing the boundary is when using thematically formatted areas like the states above a background layer in a complex map. In such cases we will often not use styles showing apparent boundary lines to prevent the map from becoming too “busy.”

At other times we will wish to emphasize the boundaries between areas to a degree even greater than is possible using area styles that color the edges of areas. To do so, we can create boundary lines using real line objects that follow the exact shape of the edges of the areas. We can then format those boundary lines as thick or thin as we would like them to be.

**Using Boundary Lines**

Manifold makes it easy to create lines that follow the edges of existing areas or to create areas that “fill” closed lines.

If we start with a group of areas, we can instantly create boundary lines for each area using objects using Transform - Boundaries command.

The new lines shown in red are new lines in addition to the "apparent" lines at the area edges, as may be seen in the middle illustration where the boundary lines have been offset slightly:
If we have separate lines that run along the boundaries of areas we can format these lines (say, by changing their color and size) to achieve a wide range of representational effects than is possible just with area styles:

We could also achieve the above visual effect by simply changing the formatting of area borders to use black foreground color and a thicker border line. In the above case we achieved the visual effect by creating new objects (using the transform tool bar operator) that are area boundary lines and then formatting them. If we simply changed the formatting of the area borders no new objects would be created although the visual effect would be the same.

**Notes**

A quick way to see if lines are in the drawing is to set the selection mode to lines only and then make a selection with the mouse. If lines are present, they will be selected and shown in red selection color. It's also a good idea to keep boundary lines in a different layer.

The antialiasing appearance of area boundaries is controlled by the Antialias Lines choice in the Tools - Options - User Interface options.
Transparent Area Styles
Maps often contain areas in drawing layers where we would like to be able to see objects in layers "below" the area. There are two ways this may be accomplished:

- Use Layer Opacity in the map layer to apply a given percentage opacity to the entire layer.
- Use a "transparent" area style to provide open space of transparent color within the area format style that allows objects below to be visible.

Both methods have advantages and disadvantages.

The advantage of layer opacity (as seen above) is that it may be applied within a map at any time to allow items underneath an area to be seen. More or less layer opacity can be applied to achieve exactly the right effect desired. At any time we can remove it since layer opacity is a property of the map, not of an individual drawing.

The disadvantage of using layer opacity is that it changes the appearance of objects. By making them transparent it can mute bright colors. Because layer opacity must be set in a map layer we could use a drawing in a different map without remembering to set layer opacity and not realize that objects beneath the layer are hidden. See the Layer Opacity topic for a full discussion of this capability.

Another way of allowing areas in drawings to reveal what is below them is to use "transparent" area styles. These are area styles that consist of hatch patterns where a large part of the pattern is either foreground or background style color and the color has been set to transparent color to allow regions beneath the area to show through.

The example above shows an area drawn with formatting that allows objects below it to be visible. To accomplish this, pick an area style that has large extents of contrasting background or foreground color and then use transparent color for one of the colors. For example, in the illustration above we chose the area style using fine dots of foreground color on a large extent of background color. We then set the foreground color to white and the background color to transparent color.
Drawings

Click on the transparent color well at the upper right of the main palette to choose transparent color.

Use of transparent color is what gives an area style a transparent effect. Any objects that are below the area will be visible through regions of transparent color. In the case of the map illustration above, all objects in layers below the area will be visible through the transparent color parts of the area. When we use the term "transparent area" we are not referring to any special sort of area or style. We are simply referring to any ordinary area that's been formatted using transparent color for foreground or background color so objects below it may be visible.

The above illustration shows the area in the topmost layer. One may also use transparent areas in drawings anywhere within the layer stack.

For example, this illustration shows the same area where the layer tab for the layer containing the area has been dragged to the right so that it is below the drawing layers containing the green line and some of the various dots, but above the layers containing the other lines. Note how the area overlaps the green dot on its right most border, but is overlapped by the green, blue, and purple dots within the area.

More than one area can use transparent color with various styles to appear transparent. The illustration above shows a second area that partially overlaps the first area. The new area was created in a drawing layer above the white dotted area, so the foreground parts of the new area will be drawn over the white dotted area.

The examples employ area styles where the background color makes up most of the pattern and so transparent color has been used for background color. Manifold has other area styles where the foreground color is the predominant part. For such area styles, one can use transparent color for foreground color.

In addition to areas, we can always use transparent color with point and line styles (usually the background color) to achieve partial transparency of points and lines.

This topic has discussed transparent areas as used in separate areas. However, one often encounters overlapping areas drawn within the same drawing layer, which can result in ambiguous visual effects. See the next topic, Overlapping Objects for a discussion of issues that may arise and for notes on why use of transparent area styles can help resolve ambiguities in such cases.
Tech Tip

When an area's foreground color is set to transparent color the entire area will not appear. This convention works with points and lines as well and is a useful way of turning off some objects in a drawing. For example, if a thematic format is defined that uses the Selection (I) intrinsic field to show foreground color as transparent if an object is not selected and black if an object is selected, then only selected objects will appear.
Overlapping Objects
Maps will often contain objects that overlap. Lines representing streams might run over lines that represent roads, and both sets of lines may be drawn over areas. In Manifold, the best way to control what objects are drawn over other objects is to keep them in separate layers (that is, separate drawings) and to arrange the layers in the order desired within a map.

When objects are all in the same layer they are drawn in the following order from uppermost to lowest:

- Points
- Lines
- Areas

This results in a natural display for most geographic maps. Within the same type of objects (such as overlapping areas in the same layer, for example) the order in which each individual object is drawn on screen depends in unpredictable fashion on their occurrence of data within the drawing. For this reason, it is not a good idea to have overlapping objects such as areas in the same layer or drawing. The correct method of specifying display order is to place objects into different layers and to then order those layers as desired within a map.

There are numerous methods within Manifold of selecting different sorts of objects so that they may be placed into different layers. For example, Objects that are co-mingled in the same layer often have different data attributes that may be used to separate them. We can use SQL or column sorting in Table View to select all lines and points with values in a "Roads" data attribute and then send them off to a new drawing.

There are rarely ambiguities involved when lines and points are organized in layers. It's easy to take care of obvious issues such as making sure that points are located in drawing layers above areas that would otherwise hide the points. Layers that contain overlapping areas within the same drawing are more complex to manage because of visual ambiguities that might arise if we organize them and format them carelessly.

Overlapping Areas
Although the correct way of ordering which objects are drawn over others is to use layers, one might encounter maps that have yet to be organized into orderly layers. This happens when importing maps from foreign formats or when drawing maps casually. At times one might launch into a project with casual expectations and then after some work realize that a drawing has become complex and needs to be better ordered. Areas are especially likely to overlap in unpleasant ways when importing and combining drawings from many different sources.

When overlapping areas occur in the same layer, it is possible to encounter confusing effects. The same rich repertoire of formatting effects that enables Manifold to provide almost any desired display can also mislead a user if applied carelessly. This discussion explores some effects one should keep in mind when working with areas that might overlap.

First, a reminder of the effects of "transparent" area styles:

Our first example shows an area that's been formatted with a simple crosshatched area style. If this is the only area in the map, there's no issue. However, if the area is drawn on a white background, it is not clear from looking at the area if the "holes" in the crosshatch mesh are filled with white background color, or with None background color. Both cases will have the same appearance when white is also used for the background color for the entire map.
By changing the map’s background color to light brown we can see the difference between the two cases. The example above uses a white background color within the crosshatch area style, while the example below uses a None background color for the crosshatch area style.

Note that using a white background color with the crosshatch area style "fills in" the "holes" in the crosshatch mesh with solid white color. Anything below the area will not be visible through the solid white color. Using None leaves the holes in the mesh transparent, so that anything below will be visible through the mesh.

If two areas overlap in the map one may encounter various effects depending on how the areas have been formatted. The example above shows two areas in different drawing layers within the same map where one area is formatted in a solid color and the other area has been formatted with a crosshatch style using None as the area background color. Because we used a transparent area style for the crosshatched area it is clear that the areas overlap, but only because the transparent area appears in a layer above the blue area.

It is wise to format potentially overlapping areas so that users of the map can instantly see any overlaps without needing to check layer ordering or otherwise resolve ambiguities. For example, the above illustration might be unambiguous if we know that the blue area is above the hatched area. Suppose we’re not sure if a transparent area style were used? It could well be that the situation above is an example of two areas overlapping or simply of a triangular blue area fitting perfectly without overlap into a crosshatched area shaped like this:
The reverse situation is also ambiguous:

Again, in this case if we were unaware that the area was formatted using a crosshatch style background color of white we could not tell whether it is simply drawn as overlapping a blue triangle area, or if it fits perfectly into an irregular blue area shaped as follows:

To be sure, we would have to check the layer ordering and the formatting used for areas.

For these reasons, we suggest using the following guidelines when formatting areas:

- Do not use the same background or foreground color within area styles as is used for the map background, or as occurs as a widespread color in lower layers. This allows areas to stand out even if “open” area styles like the crosshatch style are used.

- If overlaps might be an issue, use None as the predominant foreground or background color in “open” area styles like the crosshatch style. This assures that ambiguities in possible overlapping / non-overlapping situations will be easily resolved because overlapped areas will be visible through the None parts of the area style regardless of layer ordering. At times, however, one might wish to use a solid background color even within open styles to distinguish areas from the background. The need to do so should be balanced against the need to unambiguously portray overlapping areas.

- Avoid careless mixing of solid and open mesh area styles in the same map. Although one may always produce an example such as that below that is for the most part unambiguous, in complex real-life maps the situation is not always so clear. Using None in open styles will resolve ambiguities when the open styled area is drawn over a solid style, but if the solid style happens to be drawn over the open mesh style an ambiguity remains.

**Overlapping Area Effects when both Areas use Open Styles**

It is tempting to simply advise the user to always use an open area style (such as cross hatch) together with None background color for all areas in the a map whenever there is a possibility that areas in the same map may overlap each other. In general, this is not a bad idea, but it does have one nuance of which the reader should be aware.
When two overlapping areas are formatted in the same color and cross hatch style as shown above, the crosshatch pattern is continued uniformly throughout both areas. The "border" line typical of open mesh area styles will be drawn for both objects and so will provide clarity that there are two areas involved.

The effect when the "lower" area in the same drawing is selected shows the highlighted border. Note that even though the portion of the crosshatch pattern associated with the "upper" area is not displayed in red selection color the situation is still clear.

Perhaps the best refinement when overlapping areas occur in the same map would be to format areas in each drawing layer using a slightly contrasting color and slightly different open pattern area style as shown above.

**A "Real Life" Example**

Manifold Technical Support received the following question from a Manifold user:

"Suppose you have Area Z. Completely contained in Area Z is Area Y. There are points inside Y. There are points inside Z that are not inside Y. When the query to show all points within Z is given, all points, both inside Z and Y are selected. I don't know if that is intended."

The resolution was a matter of visual ambiguity caused by overlapping areas.
Suppose the illustration above shows the situation as described by the customer showing the points that are inside area \( Y \). Based on the visual appearance only, there is some ambiguity as to whether those points are also inside area \( Z \) as well.

If we move area \( Y \) out of the way, we might discover that area \( Z \) may or may not have a square shaped hole in it. The illustrations above show the two possibilities.

If \( Z \) does not have a hole, then the dots in the original illustration were clearly **Within \( Z \)** at the same time that they were also **Within \( Y \)**.

Using transparent area styles can clear up any ambiguity right away. The example above shows that there is no hole in area \( Z \) and so the points really are within \( Z \) as well as being within \( Y \). Another way of resolving the ambiguity is to quickly click OFF and ON the layer containing area \( Y \) to see what's going on with area \( Z \).

**Resolving Visual Ambiguities with Layer Opacity**

Individual layers in maps can be set to a given percentage opacity. See the Layer Opacity topic for examples and details.
When layer opacity is used in a map it is often easy to quickly resolve visual ambiguities between overlapping objects in different layers, especially if the default white background is clicked off and the checkerboard background pattern is revealed. The screen shot above, from the Layer Opacity topic, shows two overlapping squares that are in different layers where each layer has 50% opacity. The overlapping squares are easily distinguished as different objects.

However, if both of the overlapping square areas were in the same drawing layer opacity would not help us. The two squares would first be rendered as they would be in any case in the drawing (in a visually ambiguous way) and then layer opacity within the map would be applied. The result still would be visually ambiguous, it would just be fainter.

Consider a round area drawn within a drawing so that it overlaps other areas.

If the drawing is shown in a map with 50% layer opacity the effect is applied to however the drawing is rendered in the ordinary course of events.

If the round area were drawing in a different drawing and placed in its own layer above the other objects, then if both layers were given 50% layer opacity we would get a combined layer opacity effect that allows us to see through the round area.

The conclusion is that layer opacity within maps is fine for resolving visual ambiguity when the objects involved are in different layers but that it cannot resolve visual ambiguities between overlapping areas in the same drawing.
Lines and Line Styles

Settings in the format toolbar control the appearance of lines. The format toolbar settings apply to all lines in a given drawing. The **size** parameter sets the thickness of the line in printer's points, the same unit of measure used for font sizes. One printer's point is 1/72 inch.

![Map with layers](image)

By default, lines are shown in a solid line of size 1 with black foreground color. The illustration shows a map with a layer containing lines positioned above a layer containing the background areas.

![Solid line style](image)

We can change the line style by clicking on the **Style** button for lines and choosing a different style.

![Dotted line style](image)

A dotted style shows alternating foreground and background color. Because the background color used is the same color as that used for the areas, the line appears as a dotted line of foreground color.

![Dotted line with white background](image)

We can change the background color to white to show the alternating color effect of this line style.
When size is increased in dotted line styles, only the thickness of the background color is increased. This provides a vivid effect.

If we use the Layers pane to turn on some other layers in the sample map, we can see how the vivid style helps these lines stand out from other lines. The format toolbar illustration shows the formatting for the blue lines in one of the other layers, a “hydrography” layer showing water features such as streams and lakes.

In this illustration we have increased the size of the dotted lines to 5 to give them a thicker appearance. We have also “toned down” the brightness of the lines by changing the opacity of the layer hosting them to 50%. See Layer Opacity for information on changing the transparency/opacity of layers. Making the layer partially opaque allows some of the blue lines behind the dotted line layer to show through. Clever use of opacity will often allow complex maps to be more legible.
In the illustration above we have changed the style back to a solid line, we've changed the size to 3 and we've changed the foreground color to yellow. Solid styles that are "all foreground" will increase the thickness of the foreground color as well.

**Directed Line Styles**

Some line styles are asymmetric in that their appearance depends on the direction of the line. All lines in Manifold have a "direction" that is implied by the order in which they are drawn from the first coordinate that defines the line to the last.

A triangle line style is one such style. When looking along the direction of the line from the beginning to the end the triangles point to the left in this style. This can be seen by drawing a line.

We can use this style with the Insert Line tool to draw a new line by clicking at the left and then clicking at the right in the direction shown by the red arrow.

The result is a line drawn in triangles with the triangles pointing to the left along the direction of the line.
Suppose we draw a second line in the opposite direction as indicated by the red arrow.

In this case, the triangles will point in the opposite direction.

We can see that the orientation of the lines depends upon their direction, as shown by the red arrows. To switch the direction of a line (and thus the orientation of any asymmetric line style) select the line and use the *Reverse Lines* transform toolbar operator.

**Lines with Arrowheads**

Manifold includes directed line styles that have arrowheads. The arrowheads are placed at the end of a line.

In the example above four lines are drawn using directed line styles with arrowheads. The lines are drawn in a map above a layer containing points.

Line styles can be combined with other line styles in a map. The illustration above shows a line that was copied from one drawing and pasted into another drawing. Both drawings were shown together in a map. The upper drawing formats the line using a series of arrowheads for the line with the same yellow color for foreground and background. The lower drawing formats the line as a single thick black line.

**Antialiasing**

Antialiasing is a computer technology used to make lines appear smoother and less jagged.
Without antialiasing lines are drawn using pixels where each pixel is either turned on in the color used by the line or it is turned off. The result is a jagged line appearance for some lines.

Seen in detail, the jagged appearance arises from the stair-step nature of lines drawn at some angles for which transitions from one pixel to the next must be abrupt.

Antialiasing as seen above provides a smoother appearance to lines.

Antialiasing works by interpolating pixel colors between the color specified for the line and whatever color is visible surrounding the line. In the case of a black line seen against a background of white color antialiasing will use gray scale colors between white and black in addition to the pure black of the line. When seen from afar the use of interpolated colors fools the eye into seeing a smoother line.

Manifold can draw lines either with or without antialiasing as specified in the Tools - Options - User Interface options. The default setting uses antialiasing to provide lines with a smoother appearance.

Although antialiasing usually provides a better visual effect it does have some drawbacks. Lines will appear slightly thicker with antialiasing and some line styles or lines drawn at certain angles may appear to be unacceptably fuzzy compared to other lines. We may also want to turn antialiasing off before converting a drawing to an image if the touch selection will later be used with the image to select all lines. Touch selection can be used to select pixels that represent a line in an image if the line is not antialiased because all of the pixels in the line will be the same color. Touch selection cannot be used conveniently to select all the pixels that represent a line in the case of antialiased lines because the pixels will be many different colors.

**Transparent Foreground Color in Lines**
Choosing a transparent color for foreground color for lines will turn off display of lines. This is a short-cut way of turning off lines in certain effects. This allows a thematic format to be arranged so that line foreground color is transparent with certain values, thus turning off for display any lines with transparent foreground color. For example, if a thematic format is defined that uses the Selection (I) intrinsic field to show foreground color as transparent if a line is not selected and black if a line is selected, then only selected lines will appear.

**Note**

The /slist command line option provides a handy way of getting a list of all available formatting styles and their names for use by programmers. See the Command Line Options topic.

**See Also**

Creating Bordered Lines

Thematic Formatting
**Point Styles and Sizes**

Settings in the format toolbar control the appearance of points. The format toolbar settings apply to all points in a given drawing. The **size** parameter gives the size of the point symbol in printer's points, the same unit of measure used for font sizes. One printer's point is 1/72 inch.

Point objects are shown by default as small gray circles. The settings in the format toolbar are a size of 3, a round style, a foreground color of black and a background color of light gray. The background color is the "fill" color in most styles.

To change the size we can click on the size well.

In the resulting pull-down menu we can choose the size desired. In this case, we'll choose **12**.

All points in the drawing will immediately appear as size **12** circles.

To change the style we click on the style well.

We can scroll down through the menu of styles and choose a style that shows a dollar sign.
All points in the drawing will now appear as dollar sign icons in size 12.

To change the color, we click on the background color well.

We can choose green from the palette of available colors. The More button calls up a classic Windows color-picking dialog allowing us to choose any color.

Changing the background color to green changes the "fill" color used in this particular point style. Styles appear in the style menus with dark gray indicating where foreground color will be used and light gray showing where background color will be used.

The format toolbar summarizes the settings we now have in use

Displaying Different Points

We have two methods to display different point styles, sizes and colors in the same window:

- Use a map with different layers. Each layer in a map is a different drawing, so each layer can have different formatting for points. Map layers provide a means of controlling overlaps since upper layers are drawn over lower layers. If we have one layer with many small points and another layer with a few large points, if we want the small points to never be overdrawn by the large points we can move their layer up above the layer containing the large points.

- Use thematic formatting. Thematic formats can change the appearance of points based on the contents of a data field in the drawing’s table. Thematic formats can vary the appearance of points within the same drawing. The same drawing can host a different thematic format for point style rotation, size, style, foreground color and background color.

Points Appearing in Line Drawings
Many drawings that may be downloaded from government Internet sites will have a point object positioned at the ends of all lines. For example, virtually all USGS maps that show roads or watercourses will have point objects positioned at the ends of the lines. This is done so such lines may be used within network studies. However, when such drawings are imported into Manifold the default point size of 3 will show lots of points throughout the drawing.

If we intend to use such drawings within network analytics, we should preserve the point objects. If we want to use the drawing for purely illustrative or measurement work we can remove the points.

To remove the points, we begin by setting selection modes to Select Points only.

We can then use Edit - Select by Type to select all the points in the drawing. A keyboard shortcut is to use CTRL-T to select by type.

The selected points will be highlighted in red selection color. We can then press Delete to delete them.

The result is a drawing consisting of lines only without any points.

**Single Pixel Points**

Points of size 1 appear as slightly larger than one pixel. Because a single pixel is easy to overlook, Manifold sets the default size of points to 3 so that points appear in drawings in an obvious manner. The presumption is that if a drawing contains points they are intended to be seen. Because single pixel points are so small they will blend into any lines they terminate. We can make unwanted points seem to disappear from line drawings by changing their size to 1/4.
That's fine for a quick fix in some cases, but as soon as the formatting of the lines changes the points may become visible again. Consider the following set of illustrations as an example.

If we begin with a typical drawing of roads downloaded from USGS we will see points of size 3 at the ends of all lines.

We can change the size of the points to $\frac{1}{4}$ so they blend in with the lines. That's fine, but it is not a permanent solution if we intend to change the formatting of lines.

If we change the color of the lines, we will have to also change the color of the points from the default black foreground color so they do not appear as seen above.

Selecting the points and deleting them removes them from a drawing.

**Custom Point Styles**

Manifold can load point styles from four sources: point styles that are compiled into the product, point styles that are loaded from graphics files, point styles that are loaded from True Type fonts installed in Windows and point styles created using custom scalable styles that are provided by the user. Point styles are loaded from graphics files or from True Type fonts based upon directions specified within `.xml` files found in the `Config` folder.

Point styles created from True Type fonts may be used just like any point style, except that they are always created using one color, foreground color, only. Many of the point styles that appear in formatting menus are taken from the Microsoft *Wingdings* font and can be formatted in foreground color only. Point styles created from graphics files cannot have their colors altered, but they can use any color (as specified in the graphics file) for their pixels.

See the Customization and Custom Point Styles and the Custom Scalable Styles for Points, Lines, Areas and Labels topics for additional information, including instructions for adding your own point styles.

**Transparent Foreground Color in Points**
Choosing a transparent color for foreground color for points will turn off display of points. This is a short-cut way of turning off points in certain effects. This allows a thematic format to be arranged so that point foreground color is transparent with certain values, thus turning off for display any points with transparent foreground color. For example, if a thematic format is defined that uses the Selection ($I$) intrinsic field to show foreground color as transparent if a point is not selected and black if a point is selected, then only selected points will appear.

**Note**

The `/slist` command line option provides a handy way of getting a list of all available formatting styles and their names for use by programmers. See the Command Line Options topic.

**See Also**

- Rotating Point Styles
- Thematic Formatting
Thematic Formatting

Thematic formatting is the process of automatically coloring objects in drawings based on the value of data fields in tables associated with the drawing. This capability is called "thematic mapping" in some GIS systems. Manifold refers to the process as thematic formatting since it is simply changing the formatting of objects in a drawing or a map based on the value of a data field.

Manifold can use thematic formatting to automatically change:

- Foreground color
- Background color
- Style
- Size
- Rotation (for point styles and labels)

All four of the above formatting characteristics may be simultaneously changed via thematic formatting at the same time for areas, area borders, lines or points. In one drawing, for example, the foreground color of areas may be changed using one field while the style of areas may be changed using a different field. In the same drawing the style and size of points may be changed using other fields.

Uses for Thematic Formatting

Change Foreground or Background Color: Color adjacent areas in maps with different colors so they more obviously stand apart. This effect uses the Color dialog for drawings.

Change Size: Make points larger or smaller automatically to show larger or smaller populations, profit, number of customers or other data.

Change Color: Color countries, provinces or other regions to show differences in population or other data fields.
**Change Point Style:** Change point styles to show different characteristics at different locations. See Custom Point Styles for information on adding point styles created from images like those seen above.

**Change Area Style:** Change area styles to set areas apart or show trends with different patterns.

**Change Size:** Make lines thicker or thinner to show differences in data such as traffic volume associated with each line.

All of the above illustrations are the same map and the same data with different thematic formats.

Thematic formatting chooses a data field associated with the drawing and then assigns all the objects being formatted into the desired number of intervals based on that field. Objects can be assigned to intervals using several different methods. We can then specify how each interval should be formatted.

**To Create a Thematic Format**

1. Choose the drawing to be formatted by clicking it open or by clicking on its layer in a map.
2. In the Format toolbar, click on the display characteristic to be formatted: for areas, lines or points choose the foreground color, background color, style or size.
3. In the pull-down choice dialog that appears, choose Theme.
4. In the thematic formatting dialog, choose the controlling Field to be used.
5. Choose the Method to be used to construct intervals. Equal Count, for example, will assign approximately the same number of objects to each interval.
6. Choose the number of Breaks between intervals. This specifies the number of intervals as well.
7. Change the Align to value if even numbers of tens, hundreds, thousands, etc., values are desired for intervals. Change the Range if the default entire range needs to be extended or contracted.
8. Press the Tally button to create the given number of intervals. If interval numbers different than those created by the Method are desired, double-click into the interval numbers to change them.
9. Either use a preset color palette or click into the interval color boxes to change colors to whatever range is desired. To use a palette, select it in the Palette box and press Apply to apply the preset to
the intervals. Check the **Preview** box to try out different combinations and see how they look in the drawing without committing the changes.

10. When satisfied, press **OK**.

When a formatting characteristic, such as foreground or background color, style, size or rotation has been thematically formatted the corresponding color or sample well in the format toolbar will change appearance to show that characteristic is now specified by a thematic format.

Color samples, like the two different examples shown, will attempt to show the range of colors used in the thematic format. Obviously, there is not enough room in a small toolbar sample to show many colors, so not all colors in a complex format will appear. The idea is to provide enough colors to refresh our memory of the thematic format specified.

Characteristics such as style or size will be represented by the thematic format icon. When we see this icon in a format toolbar we know that the corresponding formatting characteristic is specified by a thematic format.

**Thematic Format Dialog**

The thematic formatting dialog is used to apply colors, styles and sizes automatically based upon the contents of a controlling field. Not all controls will be enabled for all types of formatting. For example, when changing styles or sizes the **Interpolate**, **Lighten**, **Darken** and **Grayscale** controls will not be enabled.

In the **Values** pane the numbers at left are the numeric breaks between intervals. The color wells at right show the coloration that will be applied to each interval. The small numbers immediately to the left of the boxes show how many objects fall into that interval. In the example above we thematically format provinces in Mexico by their populations using the **Spectrum** palette. The **Natural Breaks** method has assigned 7 provinces to the interval containing provinces with populations less than 580000 and only 3 provinces occur in the interval from 580000 to 1051000. There are 11 provinces in the next interval and so on.
The (default) color well shows the color that will be used by default for objects that are newly created that do not fit into any of the existing thematic categories. This is most important in cases where a thematic format has been created based upon individual values and then, after the format has been created, a new value is introduced when a new object is created.

**Thematic Format Dialog Controls**

**Field**
The data field that controls the thematic format. All data fields available with this drawing will be available in this list box. Also available will be intrinsic fields that are automatically computed by the system.

**Method**
The method used to classify records into different intervals.

**Palette**
Preset color combinations that may be applied to intervals. Press **Apply** to apply the palette to the selected intervals. Palettes are scalable and will be interpolated to apply the color range to a greater or lesser number of intervals.

- **Apply** - Apply the chosen palette to the **Values** pane. This allows scrolling through the palettes without changing colors until we press **Apply**. Pressing **Apply** only changes the color scheme in use for values. It does not change the thematic formatting of the drawing until the **OK** button is pressed. To see a preview of how the applied colors will look, use the **Preview** check box.

- **Reverse** - Reverse the formats used in the values box from high to low.

- **Interpolate** - Change the colors or sizes used in the values boxes by interpolating between the top and the bottom boxes. A quick way of creating smooth gradients of colors or sizes.

- **Lighten** - Lighten all colors. Each click on the Lbutton lightens the colors a bit more.

- **Darken** - Darken all colors. Each click on the button darkens the colors a bit more.

- **Grayscale** - Convert all colors to grayscale.

- **Move Up** - Enabled when the **Unique Values** method is used and an interval (value) has been selected. Move this value up in the range of intervals shown.

- **Move Down** - Enabled when the **Unique Values** method is used and an interval (value) has been selected. Move this value down in the range of intervals shown.

- **Load from File** - Load a previously saved theme from an XML file. Works with all field types except lookup fields. See the Custom Palettes and Themes topic for information on how the XML file is structured.

- **Save to File** - Save this theme to an XML file. Works with all field types except lookup fields. See the Custom Palettes and Themes topic for information on how the XML file is structured.

**Values**
A display of intervals created using the specified **Method**. The **Values** pane is updated with each press of the **Tally** button. If a **Palette** is used, the color wells will be updated with each press of the **Apply** button. The values may also be changed by double-clicking into any value to edit the number. The (default) color well allows editing the default value that will be applied to any objects outside the thematic format specified, such as, for example, if a new object is added with a value outside the
range of a range of **Unique Values** that have been specified.

**(interval numbers in the Values pane)** Double-click on an interval number to set it manually. For a different arrangement of intervals, choose the desired **Method** and the desired number of **Breaks** and press **Tally**.

**(color wells in the Values pane)** Double-click on a color, style or size sample to change it. Choosing a **Palette** and pressing **Apply** will apply the palette colors to the **Values** pane's color wells. Press **Interpolate** to create a smooth gradient of color from the topmost well to the lowest. Press **Reverse** to reverse the order of colors.

Continuous Shading Interpolate colors shown in the wells so that values between the numbers shown for the intervals will result in color shades that are taken from a continuous gradient of color.

**Align to** Select the number of digits to align **Values** to even values of tens, hundreds, thousands, etc.

**Range** Shows the lowest and highest values that occur in the controlling field. Change the values to specify the range over which intervals will be tallied. Values outside the range will be assigned to uppermost and lower intervals that bracket the intervals over which the method specified is tallied.

**Reset** Reset the **Range** to the lowest and highest values that occur in the controlling field.

**Breaks** The number of breakpoints between intervals.

**Tally** Re-compute intervals by specified method using given number of breakpoints.

**Preview** Temporarily shows the effect of the format when **Apply** is pressed.

**Methods**

- **Equal Count** Assign interval values so that each interval contains the same number of objects.
- **Equal Intervals** Assign interval values so that each interval contains the same range of values.
- **Exponential Intervals** Assign interval values so that each interval contains an exponentially increasing number of values.
- **Natural Breaks** Find clusters or groupings of objects by the given field and assign break values so that each cluster is in a different interval.
- **Standard Intervals** Choose break values so that each interval represents one standard deviation.
- **Unique Values** Used with text and enumerated fields. Assign a break value for each unique value that occurs in the field.

**Thematic Formatting Capabilities**

Thematic formatting uses data fields to independently control:

- Foreground color
- Background color
The above characteristics may be thematically formatted independently for areas, lines and points. In effect, it is as if we can set twelve different thematic formats for each drawing (four different types of formatting characteristics times three types of objects results in twelve different thematic formats available per drawing). If desired, we can use one data field to vary the size of points, another data field to vary the color of the points and yet a third field to vary the style of point used.

For example, we can increase or decrease the size of points representing cities based on the value of their population fields. The greater the population, the larger the point.

The illustration above shows cities in Texas where each city has a population field in the associated table. In the default format all points are the same size.

If we click on the **Size** button for points in the format toolbar and then choose **Theme** we can alter the size by the population field. Manifold will draw thematically formatted points by drawing the largest points first and then the next largest and so on to the smallest so that the smaller points are not covered by the larger points.

If we click on the **Background Color** button for points in the format toolbar and choose **Theme** we can also alter the background color of the points by their population. This results in a map where both the size of the point and the color of the point are changed depending on the population.
If desired, we can change **Style** with a thematic format. This is handy when preparing maps for black and white printers. **Style** can also be used to automatically vary point and line styles as well as area styles.

All of the different characteristics at once can be thematically formatted. The above illustration shows foreground color, background color and style thematically formatted. To assure that maps do not confuse the viewer it is usually wise to use the same data field in the same manner for all thematic formats applied to the same class of objects. That is, if thematically formatting the background color in areas based on a state's population, refrain from simultaneously thematically formatting the foreground color using the number of houses in the state.

**The Controlling Field**

Thematic formatting works with drawings that have a table associated with them. The thematic format is controlled by the value of a specified field in the table. This field is most often a numeric field, but it can be a text or other type field as well. Intrinsic fields will appear in the list box as available choices for controlling fields. Intrinsic fields are automatically computed by Manifold for each object in the drawing.

Intervals are created and objects are assigned to them using the given method. For example, the intervals can be created with breakpoint numbers so that each interval ends up with an equal number of objects. This is a useful method to use if we wish to avoid a situation where there are six intervals / color settings but all the objects end up in only one interval so our map is colored with only one color.

If the controlling field is a text field the only choice available for methods is **Unique Values**

**Breaks and Intervals**

Thematic formatting divides the objects into **intervals** that are set by the number of **breaks**. Each break is a value that specifies between the border between two intervals. The minimum value and the maximum values are always taken from the lowest and highest value in the data set. The break point values specify how the range from minimum to maximum is to be split up into intervals.

The following examples use a map of the United States that's linked to a table that has a series of fields giving educational achievement for each state. We will use a field that gives the percentage of the population that has advanced degrees (that is, some graduate degree beyond college).
If we specify three breaks as shown above we end up specifying four intervals. The first interval is the range from the minimum value in the table (the red box) to the first break point (at about 7.63%). The red color will be used to color the first interval. The blue color will be used to color the second interval starting at the blue break point of 7.63% and so on.

**Methods: Automatically Choose Break Values**

Once we choose the number of breaks we want we can either have Manifold choose the values for each break point or we can set the value for each break manually. Manifold has several different methods it can use to automatically choose values for each break.

The numbers for the three breaks shown above were computed by Manifold using the **Equal Intervals** method. This method chooses numbers for the breaks such that the total range from minimum to maximum value for the controlling field is split into intervals of equal size. In this example, because we have four intervals each equal interval represents one quartile of the distribution of values in the range from minimum to maximum.

The red box shows the interval from the minimum found in the data set (in this case, about 4.45% of the population) to the first break at about 7.63%. The minimum, red color is used to color states from the minimum to the first break. From 7.63% onward the blue color specified at the 7.63% break is used to color the interval. The green color specified for the second break takes over at 10.82% and is used to color objects until we get to the third break, after which yellow will be used.

Note that the **Equal Intervals** method makes the intervals the same size: that is, the range from 7.63 to 10.82 is the same size as from 10.82 to 14.01. In this example, because there are four intervals each equal interval represents one quartile. The number of objects in each interval can be quite different. In the above map, there are very many states with educational attainment in the lowest interval. There is only one “state,” the District of Columbia in the last interval. Yellow does not appear in the above map because the District of Columbia is too small to be visible at the zoom level shown.

**Applying Palettes**

The **Palette** box contains color schemes that may be applied to color wells in the Values pane.
The default theme uses default color for all intervals.

To apply a palette, choose it in the **Palette** box and press **Apply**. In the example above we have chosen the **Savannah** palette and will now press the **Apply** button.

Colors will be applied to the interval color wells as seen above. Palettes will usually have a wide range of colors available. Only some colors from the palette will be used if the number of intervals is small.

To change the order in which colors are used we can press the **Reverse** button.
The Reverse button reorders the colors so that the uppermost color is now the lowest and vice versa. Note how the blue color well is now at the bottom of the range.

We can also use the Interpolate button to rapidly create a gradient of colors from white to blue.

Interpolate is a handy way of creating gradients. It creates a smooth range of color from the top box to the bottom box. A rapid way of creating custom gradients is to double-click into the top box and choose a color and then choosing a color for the bottom box and pressing Interpolate. The Interpolate button works for colors and size gradients but does not work for styles.

Rounding Intervals with Align to

We often would like to have intervals that begin and end at even number values. This can be accomplished with the Align to setting.

To align intervals to even thousands, for example, we choose 3 digits in the Align to box and then press Tally.
The interval numbers will be changed to even thousands and the objects in each interval will be re-computed using the method assigned, as adjusted for the aligned values of the intervals.

Very important: Re-tallying intervals, even by just aligning them to even numbers, requires a re-application of the color intervals. For example, the illustration above was created by reapplying Interpolate after changing the Align to value. The usual workflow is to create the intervals desired (including any Range, Breaks and Align to options) and then apply colors.

Lighten and Darken

Some of the standard palettes provided may result in garish colors depending on the setting. For example, the Spectrum palette provides a continuous range of rainbow colors that may be too bright in some drawings.

Use the Lighten button to automatically lighten colors.

Use the Darken button to darken colors.

Using Range

An important use of Range is to assure that two drawings that appear together in a map have the same formatting over the same range of values. By default, Manifold takes the lowest and the highest values that occur in a drawing and uses those values to compute methods. Using Range can override these values.

Suppose for example we have a drawing of counties in California with a percent achievement scale on a standardized test where each county has a value on a scale from 0% to 100%. Suppose we also have a drawing for Oregon with results from the same test.

If the California drawing has test results that range from 5% to 85% and we thematically format it using Equal Intervals into 4 breaks the default formatting calculation will use the range 5 to 85. Using this range will create 4 intervals of 5 to 25, 25 to 45, 45 to 65 and 65 to 85.

If the Oregon drawing has test results that range from 10% to 90% and we thematically format it using Equal Intervals into 4 breaks the default formatting calculation will use the range 10 to 90. Using this range will create 4 intervals of 10 to 30, 30 to 50, 50 to 70 and 70 to 90.
If we use the same colors for both California and Oregon, counties with a test result of 27% will have different colors in the California and Oregon drawings, providing an inconsistent visual presentation if the two drawings are shown together in a map. We can avoid this inconsistency by telling Manifold to use a **Range** of 0 to 100 when thematically formatting the California and Oregon drawings.

If we use a **Range** of 0 to 100 and **Equal Intervals** with 4 breaks then each break will be the same in both drawings. Each drawing will have 4 intervals of 0 to 25, 25 to 50, 50 to 75 and 75 to 100 and all counties in both drawings will be colored using the same scheme.

Another use for **Range** is to exclude unusually high or low values from the main thematic coloring. For example if we have a lot of data that is evenly dispersed through the range 30 to 80 and but a handful of scattered values from 0 to 30 and from 80 to 100, we might tell Manifold to use an **Equal Count** method using a **Range** of 30 to 80. Intervals will be assigned so the bulk of data is divided into even intervals between 30 and 80 with the scattered values in the lowest and the highest intervals. The lowest and highest intervals can then be clicked by hand and changed to less prominent colors.

**Continuous Shading**

If we check the **Continuous shading** box, the color for each state will be drawn from its field value as interpolated between the colors specified at the breaks.

![Continuous Shading Image](image)

Without checking the **Continuous shading** box, all states use one of the colors specified for the intervals. Without the **Continuous shading** box checked it looks like the entire country is not highly educated with California, Colorado and New Mexico in the West and a handful of states in the Northeast providing some slightly elevated educational achievement. A very few states in the East have educated populations shown in green color.
When the **Continuous shading** box is checked, each state gets a color that is interpolated between the colors specified for each break. The **Continuous shading** box allows us to specify a relatively small number of breaks and colors yet still see some visual differences within the intervals. Using continuous shadings is the reason we have a color box available to set the color value for the "maximum" value. This box is used only when the **Continuous shading** box is checked since without continuous shadings each interval is colored with the color box of the break point that starts the interval.

With the **Continuous shading** box checked we see that the situation is more complex that it initially appeared. Most states are still in the low-achievement range with Arkansas leading the way in low educational achievement. The blue states are now seen to divide into two classes of states: lower-end states like California that are more like the low-achieving mass of states, and a few higher-end states like Colorado, Virginia, New York and Massachusetts. States in the latter group have more in common with highly educated states like Connecticut than they have with California. That they were grouped by color with California is an accident of choosing three break points to divide the entire range into four equal intervals.
A better way to reveal groupings is to use the **Natural Breaks** method. This method allows Manifold to find clusters of similar values within the field and to choose break points so that each cluster is in an interval. The above example shows **Natural Breaks** working with 5 breaks to create six intervals.

The natural breaks thematic map shows the highly irregular distribution of populations with higher education in the US. Most states fall into one of four groupings having less than 7% of the population having higher degrees. For the most part, states are clustered into different levels of mediocrity around the 5.7% level. A handful of states fall into a cluster above 8.27% and are shown in green.

The Manifold **Natural Breaks** algorithm works remarkably well. People who are familiar with these states would say that California, Washington, Illinois and New Hampshire are definitely in the same bucket when classing states by percentage of highly educated people. **Note**: the educational attainment thematic maps shown in this topic are based on Census Bureau percentages of advanced degrees in 1990.

**Interval Styles in Legends**

**Legends** will automatically adapt to the status of the **Continuous shading** box.

```
Mexico Drawing
Areas: [POBL_1990]
_<-1837981
-1837981...240047
240047..2318075
2318075..4396103
<4396103...6474131
>6474131
```

Legends for thematic formats not using continuous shading will show the range of values to which each color applies.
If the **Continuous shading** box is checked the legend will show the discrete values and colors between which a continuous range of colors is interpolated.

Automatic legends will use the same number of digits beyond the decimal point that are specified for that column in table column formatting.

**Example: Use Thematic Formatting to Show Territories**

Suppose we have a **Territory** field in our table of US States. This is a text field and contains the name of the sales territory that includes that state. Sales territories have names like "Northeast" and "West Coast".

**Step 1: Click open Thematic Format Dialog.**

Click open the drawing to be formatted, or click on the drawing's layer tab in a map to make it the active layer. Click on the format toolbar's **Background Color** well for areas. Choose **Theme** to launch the thematic formatting dialog.

**Step 2: Choose the Controlling Field and Method**

In the thematic formatting dialog, choose **Territory** as the controlling field and **Unique Values** as the method. The breaks and intervals part of the dialog will automatically change to the list of unique values found for the territory field. There are only six different strings (that is, six different territories) found in this particular example.

**Step 3: Choose Colors to be Used**

Double-click into each color well and choose a color or use one of Manifold's preset color arrangements. Click **OK**.
The resultant thematic format colors the background colors of each state area by the value found in the Territory field.

**Example: Use Thematic Formatting to Show Population**

We open a drawing of Mexico showing provinces.

The drawing's table has a field called **Population** that we would like to use to color the provinces by their populations.

<table>
<thead>
<tr>
<th>ID</th>
<th>SQMI</th>
<th>Population</th>
<th>Vl</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>26535.63</td>
<td>2003187</td>
<td>38</td>
</tr>
<tr>
<td>2</td>
<td>32089.18</td>
<td>1276323</td>
<td>23</td>
</tr>
<tr>
<td>3</td>
<td>52374.05</td>
<td>1349378</td>
<td>26</td>
</tr>
<tr>
<td>4</td>
<td>76106.63</td>
<td>1823606</td>
<td>38</td>
</tr>
<tr>
<td>5</td>
<td>23603.84</td>
<td>2264045</td>
<td>43</td>
</tr>
</tbody>
</table>

The "body" color of the provinces is set by their background color. Click on the format toolbar's **Background Color** well for areas. Choose **Theme** to launch the thematic formatting dialog.
Choose **Population** as the controlling **Field**. Choose **Equal Count** as the **Method**. This will assign approximately the same number of provinces to each population interval. Choose the **Red to Yellows** palette and press **Apply**.

Those are nice colors, but we note that the lower population intervals are colored red and higher population provinces will be colored yellow. We would like to reverse this so that provinces with higher populations will be colored red. To do so we press the **Reverse** button.
From experience we know that the very bright colors used in the Reds to Yellows palette will result in a garish map. To lighten the colors we press the Lighten button twice.

That results in a more muted palette that will still show variations in the provinces well.

The final product is a drawing of Mexico that shows provinces colored by the contents of their Population field. Provinces with higher populations are colored red. Those with lower populations are colored yellow. Right away
it is obvious that much of the population of Mexico is clustered in a small band of provinces in the middle of the country. Regions such as southern Baja California and the provinces to the East are less populated.

**Fixed Interval Palettes**

Palettes in Manifold occur in two forms:

- Relative palettes, where colors from the palette will be "stretched" and interpolated as necessary for use in the number of breaks specified in the **Format** dialog. The examples above all use relative palettes.
- Fixed palettes, where colors are associated with specified intervals. When a fixed palette is used in thematic formatting, applying the palette will automatically create as many intervals as are required by the numbers specified for the palette.

Fixed palettes are most frequently used to color surfaces that show terrain elevations. They allow a standard color scheme to be applied for specific elevations that is the same from surface to surface. Manifold includes a few fixed palettes for use with surfaces. Altitudes are predictable since the general elevation of the Earth is known and covers a fairly narrow range. Standardized palettes may therefore be created for altitudes and incorporated into Manifold.

Fixed palettes are less frequently used with thematic formatting in drawings since normally the range of values that may be displayed in thematic rendering covers a very wide range. It is usually faster to create fixed intervals by specifying the **Breaks** option, tallying the desired number of intervals and then applying a relative palette. We can then "clean up" the breaks to desired interval numbers.

However, for some applications such as land classification a specific, fixed palette may be desired for use in thematic formatting. Use the customization features noted below to create fixed palettes that have ranges of values that correspond to colors desired.

**Saving and Loading Themes**

Press the **Save** button to save a theme to an XML format file. Themes for color, size or style may be saved.

Press the **Load** button to load a theme from an XML format file. When attempting to load a theme file for a different type of theme (for example, trying to load a theme file that saved a thematic format for different point styles when working in a thematic format dialog to set color) the system will load the formatting breaks from the saved theme and will reset formatting to default.

Loading a saved theme file will also reset the **Method** to a method that makes sense for the type of field (column) being used as follows:

- When loading a file containing value-based formatting such as **Unique Values** the system sets the formatting method to **Unique Values** since this method is supported by columns of any type.
- When loading a file containing interval-based formatting such as **Equal Intervals** the system checks if the current column supports interval-based formatting (that is, if the current column is not a **Date**, **Text** or **Boolean** type field) and if the current formatting method is value-based. If so, it sets the formatting method to **Equal Intervals**.

**Customization**

Palettes used in thematic formats may be customized, and new palettes may be added to Manifold. See the Customization topic.

**The Color Dialog**

The **Color** dialog makes it easy to automatically apply a thematic format to a drawing that colors each adjacent area using a different color.
When applied to a drawing that contains adjacent areas, the color dialog will add an integer field called **Color** to the table. (If a field exists that is already named **Color** the system will create the new field using a name like **Color2**). The drawing is then examined using Manifold’s internal graph theoretic algorithms to assign a small number to each area in the **Color** field so that no two adjacent areas have the same value in their **Color** field.

The **Color** field is then used in a thematic format to specify both foreground and background color. The colors used are made slightly darker in the foreground color so that the borders of area styles will be distinct.

It’s easy to change the color scheme after applying the color dialog to automatically color areas. Simply click on the color well to be changed for areas and choose theme and then apply one of the preset color schemes. The color applied by the color dialog is simply a thematic format using the **Color** field with a **unique values** method.

Note that the word **theme** is used both for the application of a palette of colors to a drawing or a theme and is also used as the name of the **theme** component. We apologize for this minor lexicographic inconsistency, but it seems clear which is which in actual usage.

**ColorBrewer Palettes**

Users with experience in thematic formatting know that it can be sometimes difficult to choose colors for thematic formatting that do not result in a miserably garish display. Manifold includes **ColorBrewer** palettes, beginning with a **CB** in the name of the palette, which are arrangements of colors that have been cleverly picked by Cynthia Brewer to work well together. Using one of the **CB** palettes almost always results in a pleasant and legible effect. See the Miscellaneous topic and [http://www.colorbrewer.org](http://www.colorbrewer.org) for more info on the ColorBrewer project.

The **CB** palettes are each designed with a fixed number of colors, the number of colors being indicated by the number at the end of the palette name. For example, **CB Pastel 4** is a palette of four colors. The number of colors should be the same as the number of breaks used in the thematic format. So, for example, if we use the default 5 breaks when formatting a map by, say, population using the **Natural Breaks** method we should use **CB Pastel 5** as the **CB** palette so that there are five colors in use.

**See Also**

See the Color topic for fast “four-color” coloring of drawings.

**Thematic Formatting Example**

Thematic Formatting and Labels
Rotating Labels
Themes
**Thematic Formatting Example**

An example of Thematic Formatting is automatically changing the size, color or style of points based on the contents of a data field for each point.

Suppose we have a drawing with various city points. The illustration above shows our drawing of points as a layer in a map with a background map of the United States. Suppose the city points drawing's table has a **Population** field that gives the population for each city. We would like to change the size of points based on the contents of the **Population** field for each point.

![Map illustration](image)

We begin by clicking on the point size button in the format toolbar and then choosing **Theme** from the menu. This launches the thematic formatting dialog.

<table>
<thead>
<tr>
<th>Field: Population</th>
<th>Method: Equal Count</th>
</tr>
</thead>
</table>

In the thematic formatting dialog, we choose **Population** as the controlling field. For this example, we use the **Equal Count** method so that there are an equal number of points in each of the intervals defined by the thematic format. We've also used alignment to 4 digits and pressed **Tally** so that the interval breaks occur at even 10000 values.

![Thematic Format Table](image)

By default, each interval has a size of 3 for the point sizes. The other numbers show how many of our cities fall within each population bracket. We can change the point size for each interval by double clicking into each size number and changing it.
In this illustration, we’ve changed half of the values. We’ve chosen a progression of sizes that we think will be a reasonable increase in size for each interval. For example, the cities with populations from 20000 to 30000 will have a size of 4. If this turns out not to provide the desired visual effect we can open this dialog again and adjust the values.

We’ve now changed all the values. The interval from zero to the first number has a size of 1 and the interval from the last number up has a size of 10.

When we press OK the points will be redrawn with size values corresponding to the contents of their Population field. Points with larger Population will be drawn as larger size as specified in the thematic format.

Changing Color

We can also change color using a thematic format. In fact, we can change color with a thematic format even as we change size. To change color, we click on the background color well and choose Theme to launch the thematic format dialog.
In the thematic format dialog we once again choose Population as the controlling field and Equal Count as the method to use. Once again, we use an alignment to 4 digits and press Tally to get intervals of even 10,000. When creating thematic formats for colors, Manifold provides a choice of Palettes as well. Each palette provides a handy color range. We choose the Blues to Greens palette.

We press Apply to apply this palette to the intervals used.

The palette applies a range of graduated colors from blue to green to the intervals in our thematic format. We then press OK.

The points will now be drawn with color that varies by the value of their Population field. Note that the color change is in addition to the change in size for each point. This provides greater emphasis on the more populated cities.

Changing Style

If desired, we can even vary point style with a thematic format. To do so, we click on the point style button in the format toolbar and choose Theme. This launches the thematic formatting dialog for varying point style. Once again, we will use Population as the controlling field and Equal Count as the method for assigning intervals. We will also choose an alignment to 4 digits and press Tally to get intervals of even 10,000.

This time, when we double-click into each interval’s attribute we will change the style instead of changing a size value or color. The dialog opens with all circles. We can double click on the circle for each interval and change the circle to the desired style. In this case, we have specified a series of styles from a simple cross to a circle to a square and so on to an inverted triangle.

We press OK to apply the thematic format.
The points will now be drawn with a style symbol that varies in accordance with the Population field. Note that the style variation is in addition to the color and size variation specified in the other thematic formats we applied.

**Tips**

When applying more than one thematic format, it is almost always best to use the same method and the same number of intervals for all thematic formats. It's too much to ask of viewers to expect them to keep in mind that size varies in accordance with five intervals but color varies only by three intervals.

It is not easy to visualize three different variables at once even when they are each coded by size, color or style. Use multiple thematic formats with great restraint. Note how in the above examples the image with circles of varying size but all of the same color does a good job of conveying the relative populations of the towns. There is some value added in adjusting color and style in some circumstances; however, the more frequent case is that overuse of thematic formats to display many different attributes at once will lead to a confusing presentation.

A fast way to assign a range of values is to use the **Interpolate** button. Click on the topmost interval and assign the value (such as color or size) for that interval. Next, click on the bottom interval and assign the value for that interval. Finally, click the **Interpolate** button to fill in the intervals in between with interpolated colors or values.

When interpolating styles, the intermediate intervals will be filled in with the symbols that occur, in order, in the styles menu from the first interval's symbol to the last interval's symbol.

**See Also**

**Thematic Formatting**

**Thematic Formatting and Labels**
Rotating Point Styles

Point styles in Manifold may be rotated using the **Point Rotation Angle** button in the format toolbar for labels. This topic assumes the reader is familiar with formatting in general and with the techniques set forth in the Formatting Drawings and Point Styles and Sizes topics.

To rotate all of the point styles in a given drawing, open the drawing, press the **Point Rotation Angle** button and choose a rotation value.

For example, the point style shown above (a point style in the shape of a letter "A") has been rotated through 0, 30, 60, 90 and 120 degrees.

**Example**

Suppose we have a drawing of points that include a **Direction** field for each point. The points are measurements of flow direction in a stream.

By default, the points appear as small round circles.

We can change the point style to an arrow.
To rotate the points, we can press the **Point Rotation Angle** button.

In the pull-down menu we can choose **90** to set the point rotation angle to **90** degrees.

The result is that all points in the drawing will have their point style rotated to **90** degrees.

If desired, we can also rotate each point based upon the value of a field using thematic formatting. To do so, press the **Point Rotation Angle** button and then choose **Theme**. In the thematic formatting dialog that follows, choose a thematic format that sets the rotation angle by a given field, such as a **Direction** field. The fastest way to do this is to use the **Continuous shading** option in the thematic format dialog.
Using the values of a Direction field to control rotation:

1. Open the drawing, click the Point Rotation Angle button and choose Theme.
2. In the Format dialog, choose Direction as the field and any interval-based method, such as Equal Count as the method.
3. Choose 2 as the number of breaks.
4. Check the Continuous shading box.
5. Press the Tally button to establish two intervals.
6. Double click into the two Values cells and change the upper one to 0 and the lower one to 360.
7. Double click into the rotation values and change the upper one to 0 and the lower one to 360. Press OK.

The above procedure creates two intervals, of 0 and 360 degrees and then continuously formats all values in between. In effect, this assigns a continuous range of rotations from 0 to 360 degrees to any Directions values between 0 and 360.

See the Rotating Labels topic for examples using thematic formatting to set rotation angle.
In that case, each point will be rotated by the given field as seen above.

See Also

Rotating Labels
Point Styles and Sizes
Themes

Themes are project components that store formatting choices for drawings. Opening a theme shows the drawing with the formatting specified by the theme. A drawing can have many themes, each of which may show that drawing through a different set of formatting choices. Adding a new theme to a project requires virtually no storage space.

Themes may be used as though they were a drawing, for example, appearing as a layer in a map. Selections and editing commands may be done in a theme. Because a theme window is simply a drawing viewed through a particular format, any selections, edits or other changes except formatting made in any theme of a drawing will affect the drawing and all other themes of that drawing as well.

Themes are most often used with thematic formatting to show values of different data fields in a drawing. For example, the same drawing may be shown in different themes to show the values of populations, per capita income or other demographic attributes. Themes make it easy to show the same drawing in different ways without having to either reformat the drawing or make copies of the drawing for use with different formats.

To create a theme for a drawing:

1. In the project pane choose Create - Theme from the project pane toolbar or right click into the project pane and choose Create - Theme from the context menu.
2. In the Create Theme dialog choose a name and an optional description for the theme.
3. Choose the Parent drawing for the theme. This is the drawing that will appear in the theme with the specified formatting.
4. Check the Inherit formatting box if the theme is to automatically inherit any formatting changes made to the parent drawing. Uncheck this box if the theme is to have no connection to the formatting of the parent drawing.
5. Press OK.

Every theme is created based upon some parent drawing. The parent drawing appears in the theme. Any selection made in the theme will be a selection from the objects (points, lines or areas) in the parent drawing. Any editing changes in the theme, such as adding or removing objects, will be implemented in the parent drawing.

Themes may also be created based upon other themes. In that case, enter the name of the parent theme in the Parent box. If a theme is created based upon a parent theme, whatever is the ultimate parent drawing will also be the parent drawing for the newly-created theme.

Example

Suppose our project contains a drawing of Mexico provinces, created from the Mexico_eg drawing on the Manifold CD.
We can open the drawing and see that it uses default formatting.

We can create a theme by choosing **Create - Theme** in the project pane toolbar. We enter the name **Population** and choose the Mexico drawing as the parent. We uncheck the **Inherit formatting** option so that the theme takes no formatting attributes from the parent drawing.

A new theme appears in the project under its parent drawing.
If we open the theme we see it has default gray formatting.

We can click into the area background format well in the format toolbar and apply a thematic format that colors the theme based upon the population in each province, with darker greens indicating greater population. The POBL_1990 field in the example Mexico drawing gives the 1990 population for each province.
The theme will immediately show the new formatting applied.

Next, we can create another theme called **Size**.

The parent for this theme will also be our Mexico drawing. Once we uncheck the **Inherit formatting** box.

A new theme appears in the project underneath its parent drawing.
We can apply a thematic format to the **Size** theme that colors provinces based upon their size in square miles, with larger provinces having a lighter color.

We will create one more theme, called **Buses**.

As with the other themes, this theme appears in the project pane underneath its parent drawing.

We will thematically format this theme based upon the number of buses in each province.
The darker the color, the more buses in the province.

If desired, we can open all three themes as well as their parent drawing. Each theme appears like a drawing in a drawing window and each theme has all of the characteristics that a drawing has. Each theme is derived from the parent drawing. Although there may appear to be four different “drawings,” there is in fact only a single drawing and only one copy of all the objects in that drawing. The themes are just different ways of displaying that same drawing by using different formatting. We can have as many themes as we like and still there will be no increase in the amount of disk space required for the project.

**Selections and other operations**

We can make selections in a theme window, edit objects and otherwise do anything we could do in a drawing window.
For example, if we have all three themes open and their parent drawing as well, a selection made in any of the windows will immediately appear in all of the other windows.

If we make any edits in any of the windows, such as deleting objects or changing their position or size or form, the edits will take effect in all of the windows. The reason is that a parent drawing and any theme based upon that parent all show the same objects. The themes simply show those objects with different formatting.

Any selections saved in the Selections pane will also be the same for all themes of the same parent drawing.

Themes inherit any views in the Views pane from their parent drawing. Adding a view to any theme will add it to the parent drawing and to all other themes of that parent.

Likewise, themes inherit any Viewbots from their parent drawing. Adding a Viewbot to any theme will add it to the parent drawing and to all other themes of that parent.

**Themes and Toolbars**
Themes can be used in the Query and Transform toolbars. Object sets defined on themes will always use the name of the theme's parent drawing. For example, if there are selected objects in a theme named T whose parent is a drawing named D the selection choice will appear in the query or transform toolbar boxes as Selection in D.

This is done so that it is immediately clear that modifying a theme will modify the parent drawing (and thus all other themes bound to that drawing).

Inherited Formatting

In the example above we've considered a simple use of themes, where each theme is formatted completely independently of the parent drawing. Themes may also be created so that some or all of the formatting of the theme is automatically derived from the formatting of the theme's parent. Such formatting is called inherited formatting.

When a theme is created with inherited formatting, any changes made to the formatting of the parent will be applied automatically to that theme as well for those formatting characteristics that are inherited. If we like, at any time we can disinherit any particular formatting characteristic so that it is no longer inherited from the parent and can be set independently of the parent. Likewise, it is easy at any time to change a local formatting characteristic in a theme to be automatically inherited from the parent.

Inherited formatting allows us to pick and choose what formatting characteristics are automatically displayed in themes based upon changes to their parents. A theme can even use another theme as a parent, which makes it possible to create subtle and sophisticated relationships between different themes to specify what formatting characteristics will be automatically changed and what characteristics are unique to a specific theme.

Because the project pane always shows the hierarchical relationships of parent drawings to themes and then to themes that have other themes as parents, we can always use the project pane as our guide to the relationships we have created.

To create a theme with inherited formatting:

1. In the project pane choose Create - Theme from the project pane toolbar or right click into the project pane and choose Create - Theme from the context menu.
2. In the Create Theme dialog choose a name and an optional description for the theme.
3. Choose the Parent drawing for the theme. This is the drawing that will appear in the theme with the specified formatting.
4. Check the Inherit formatting box if the theme is to automatically inherit any formatting changes made to the parent drawing. Press OK. This creates the theme with all formatting characteristics being inherited from the parent.
5. Open the theme. To disinherit any formatting characteristic, click the formatting well for that characteristic and set some local format. As soon as we set a local format in the theme the inheritance for that format is disconnected.
6. If we change our minds and would like any characteristic to be inherited again, simply click on the formatting well for that characteristic and choose Parent in the pull-down color well dialog.

Example

Let's begin by creating a theme from the Mexico sample drawing (now simply called "Mexico").
However, this time we will check the Inherit formatting box.

As before a new theme appears in the project pane.

If we open the theme we see it has default formatting.

The format toolbar for this theme now shows four corner brackets displayed for each formatting characteristic. Corner brackets indicate that a particular formatting characteristic is inherited from a parent. Because this theme was created with Inherit formatting checked, all of the formatting characteristics are inherited from its parent, the Mexico drawing.
We next click open the *Mexico* drawing and thematically format the background color as seen above.

If we click open the *Theme 1* theme, as expected it has the same formatting as its parent *Mexico* drawing. The formatting changes made to the area background in the *Mexico* drawing have been inherited automatically by the *Theme 1* theme.

At any time we change an inherited formatting characteristic into a non-inherited formatting characteristic simply by applying a format to that characteristic.

For example, if we would like to change the area background color for *Theme 1* into a non-inherited, local format we can simply click on the area background well in the format toolbar.
We choose a new color for the area background…

…and the format toolbar will show the area background without corner brackets to indicate it is a local format and not inherited from a parent.

**Theme 1** will now show area background color as specified within the theme and not as inherited from its parent **Mexico** drawing. If any change is made to area background color in the **Mexico** drawing it will not be propagated to the **Theme 1** theme. However, any other changes made in formatting to the **Mexico** drawing, such as changes in area foreground color, area style, line colors, etc. will be automatically propagated into the **Theme 1** theme.

We will now create a theme based upon another theme.
In the project pane we choose Create - Theme and then in the Create Theme dialog we create a new theme called Theme 2 that use Theme 1 as the parent. Note that we’ve checked the Inherit formatting box.

Our new theme is created in the project pane below Theme 2. The theme's position in the project pane hierarchy shows that Theme 1 is its parent.

When we open the theme we see that Theme 2 appears the same as Theme 1, which is what we would expect since Theme 1 was the parent and we created the new theme with inherited formatting. The toolbar for Theme 2 shows that all formatting characteristics are inherited from a parent. Note that even though Theme 1 is the parent of Theme 2, because Mexico is the ultimate parent of Theme 1 all of the objects that are seen either in Theme 1 or Theme 2 come from the Mexico drawing.

Let's now make some formatting changes in both Mexico and in Theme 1 and see how they are propagated through the two themes we have created.
In the Mexico drawing we change the foreground color for areas to blue so that the area borders are rendered in blue (a strikingly ugly combination, given the thematic formatting of the background color for areas).

The foreground color for areas will immediately change in the Theme 1 theme, because this formatting characteristic is inherited from the Mexico drawing.

In the Theme 1 theme, let us now change the background color for areas to a light green.
If we open **Theme 2** we see that **Theme 2** has inherited the light green area background color from **Theme 1**. It has also inherited the blue area foreground color from **Theme 1** which in turn inherited the color from **Mexico**.

Suppose we change a formatting characteristic so that it is not inherited but then at some future point we would like to change it back to inherited formatting? That’s easy to change.

Let’s go back to **Theme 1**, seen above. Note that the area background color well in the format toolbar has no brackets, so it is not inherited from **Mexico**.

To change it back we click on the area background color well…
…and choose Parent within the color choice dialog.

This changes the area background color into an inherited formatting characteristic and automatically Theme 1 will be displayed using the thematic format used by the Mexico drawing for area background color. Theme 2 (not illustrated) will also have the same appearance because Theme 2 inherits all formatting characteristics from Theme 1.

Notes

Although there may seem to be no downside to using themes, there is one minor user interface issue of which users should be aware. In earlier editions of Manifold System, when themes were not yet available, a user could tell from the visual appearance of a drawing layer exactly which drawing was being manipulated because every window that showed a particular drawing always showed that drawing in the same format, the drawings’ format.

Because themes make it possible to show the same drawing using many different visual appearances we should take care to use formatting in sensible ways to avoid visual confusion.

For example, suppose we have two drawings that show European countries: Drawing A is formatted with mostly green colors and drawing B is formatted using mostly blue colors. It would be unwise to create a theme A1 for drawing A that is formatted with blue colors so it looks like drawing B, and it would be unwise to create a theme B1 for drawing B that is formatted with green colors so that it looks like drawing A. We don't want users to make edits within A1 thinking they are editing the B drawing because of the visual similarity between the two.

It's true that at any moment a user could look at the title bar of a theme window or the name of a layer in a map window and compare them with the names of drawings and themes in the project pane. Nonetheless it makes sense to keep in mind the possibility of visual confusion when working with large, complex projects that might include dozens or hundreds of components.
A good approach is to adopt naming conventions that clearly state the parentage of themes. For example, we might call a drawing "US States" and name its themes "US States - Income," "US States - Birthrate," "US States - Population" and so on based upon how the themes are formatted.

**Tech Tip**

When pasting a theme as a labels component to create a bound labels component, the parent of the labels component will be the parent drawing of the theme.

**See Also**

* Formatting Drawings

**Editing Drawings**

**Editing Drawings**

There are two classes of tasks when editing drawings: creating new objects or modifying existing objects.

We can create **new** objects by:

- Importing them from a non-Manifold source.
- Pasting them into this drawing from a different component.
- Creating them as the result of a command or a script.
- Drawing them "free hand" using interactive editing capabilities as shown in Adding Shapes or in Adding Points, Lines and Areas or in the Dialog Mode and Visual Tools topics.
- Creating them automatically from images using tracing (vectorizing) capabilities.
- Using commands such as Dissolve.
- Using the Transform toolbar to create one or more objects.

We can change the size, shape and position of **existing** objects by:

- Using the Transform toolbar to change many objects at once throughout the drawing.
- Using Cut, Copy and Paste to move objects between drawings and drawing layers in maps.
- Changing the objects as the result of an editing command such as the Dissolve command.
- Editing objects such as by changing their shapes using simple interactive commands or by using the Edit Toolbar during interactive editing. If not already displayed, the edit toolbar will appear when an object is selected for editing with a **CTRL-ALT** clicking the object.
- Altering objects "free hand" using interactive editing capabilities such as using the Context Menu commands.
- Editing the coordinates that define objects using the Object Coordinates dialog.
- Editing the position of the object by editing intrinsic fields in a table.
- Changing the position of points or repositioning other objects by editing their intrinsic fields in a table.
- Moving objects in drawings using the alignment toolbar to align them as desired.
- Using Topology Factory (if the Business Tools extension is installed) to detect and correct a variety of topological errors, such as undershoots or dangles.
- Using Topology Overlay to edit objects and their data attributes in a target drawing using the areas in a source drawing.

In addition to editing the **geometry** of objects (that is, changing their location or shape) we can also edit the data fields, if any, associated with those objects by:

- Editing the table associated with the drawing.
- Using Spatial Overlay to transfer data between objects in different drawings.
- Using Transfer Heights (if available) to transfer data from a surface to the objects.
- Using the Color or Districts tools to change fields used to format or organize objects.
Most operations will automatically apply only to the selection, if one is present. If no selection is present they will apply to the entire drawing.

**Transfer Rules**

Many editing / transformation operations in drawings (see Transform Toolbar - Drawings and Drawings - Dissolve, for example) will create new objects from existing objects. When new objects are created from existing ones the system must know how to transfer data attribute fields from the existing objects into the new objects. The Transfer Rules dialog in tables sets forth the rules to be used for each particular column.
Using Cut, Copy and Paste in Drawings

Use the standard Windows Clipboard **Cut**, **Copy** and **Paste** commands in Manifold to delete objects and to move them between drawings and drawing layers in maps. There are three paths to using clipboard commands:

- **Use commands from the Edit menu.** In this case **Cut** and **Copy** will apply to all selected objects in the active drawing window.
- **Use standard Windows keyboard shortcuts:** **CTRL X** to **Cut**, **CTRL C** to **Copy**, and **CTRL V** to **Paste**. Keyboard clipboard commands will apply to all selected objects in the active drawing.
- **Right click** onto an object in a drawing and choose **Cut** or **Copy** from the drawings context menu, or **right click** anywhere in the drawing and choose **Paste** or **Paste Append**.

Windows Clipboard commands will work between two different Manifold sessions if you have two Manifold programs launched at the same time on your computer. We can copy objects from one Manifold session, switch to the other Manifold session using the Windows taskbar and then paste objects into a drawing in the new session. This is a handy way of copying or moving objects between different projects.

When copying and pasting, objects remember their geographic location. They always appear at the same geographic location no matter how the target drawing or map is panned or zoomed.

**Clipboard Commands in Maps**

When used in maps, Clipboard commands operate on the active layer only. This provides a handy way of rapidly moving objects between drawings that appear as layer in maps. Click on the source layer tab to activate it, **Cut** desired objects, click on the destination layer tab to activate it and then **Paste**.

**ALT Key in Maps**

Selection and Clipboard operations in maps work only on the active layer. To extend the action of these operations to all visible layers, press the **ALT** key. For example, drawing a selection box in a map will only select objects in the active layer. Holding the **ALT** key while drawing a selection box will select objects from all visible layers. Pressing **Delete** deletes selected objects in the active layer only. Pressing **ALT-Delete** deletes selected objects in all visible layers. Think of **ALT** as a mnemonic for "All."

**Cutting**

Suppose we have one drawing that contains a triangular area. This drawing is formatted so that areas appear in blue.

We have another drawing that contains a square and a circular area. This drawing is formatted so that areas appear in yellow.
If we place the two drawings together in a map with the yellow drawing in a layer above the blue drawing we can see the relative positions occupied by the objects in the two drawings.

Let's select the yellow square. If we use **Edit - Cut** (or the standard Windows keyboard shortcut, **CTRL x**) it disappears.

**Pasting**

In Windows, how a **Paste** command works depends on whether or not something is selected. If nothing is selected, the objects in the Clipboard are pasted as new items in addition to any objects in the target drawing. If some objects in the target drawing are selected at the time of **Paste**, the clipboard is pasted *in replacement* of those items. The selected items are deleted and then the new items are inserted.

We can see this by continuing the above example by pasting the clipboard into the blue drawing layer. The screen shots below show two different cases: on the left we have not selected anything before the **Paste** and on the right we have selected the triangle.

Suppose we have just **Cut** the yellow square. We are about to **Paste** it. On the left, nothing is selected. In the example on the right, the blue triangle has been selected.

Now we **Paste** the clipboard into the blue layer. On the left we see that the square appears in the blue layer (in blue formatting, of course) in addition to whatever is in that layer. In the example on the right where the triangle had been selected, we see that the **Paste** command replaces the selected triangle with the new, blue square. Note that if there had been many objects selected in the blue layer in addition to the triangle, *all* of them would have been deleted before the new, blue square was pasted.

As a conceptual analogy, this is exactly how Windows cut and paste works in standard Windows applications such as Microsoft Word. If we highlight a phrase in Word and then **Cut** it we copy the phrase to the clipboard. If we paste this phrase back into the document what happens depends on whether any text is subsequently highlighted. If no text is highlighted, the phrase that was cut will be pasted back into the document in addition to all existing text. If, on the other hand, we highlight some text and then do a **Paste**, the pasted phrase will replace whatever text was highlighted.

**Paste Append** is like **Paste** except that it will not delete and replace any previously selected objects.

**A Geographic Example**
Suppose we have two drawings showing features in the San Francisco Bay region. The drawing on the left shows features from the hydrography layer of a USGS map. The drawing on the right shows Congressional districts for the 106th Congress as areas. We have selected one of the districts and have used Edit - Copy to copy it to the clipboard.

If we now click on the bay_hydro drawing to switch focus to that drawing and choose Edit - Paste the area object in the clipboard will be pasted into that drawing. We have used the Format toolbar to specify use of yellow color for areas in bay_hydro, the same light yellow color used for areas in the districts map. Pasted objects will take on whatever color is defined for objects of that type in the destination drawing.

Suppose now we select the area object just pasted into the bay_hydro drawing. We open a third drawing, called California, which shows California counties as areas. We select San Mateo county in the California drawing and use Edit - Copy to copy it to the clipboard.
If we now click on bay_hydro again and Paste, the previously selected district object will be deleted and replaced by the San Mateo county object being pasted in from the clipboard.

Notes:

- The illustration above shows the pasted object in light green. When it was first pasted into bay_hydro it appeared in the light yellow color defined for areas in that drawing. We used the Format toolbar to change the color to light green. We did this to visually emphasize the replacement of the district object by the county object.

- When objects are pasted into a destination layer they are selected. This is just like pasting text into Microsoft Word: the pasted text is highlighted after it appears in the document.

A Troubleshooting Example

If we Copy selected objects like those at the left and Paste them into a different drawing and get the result at the right, the problem is that the drawing on the right uses Orthographic projection while the drawing at left uses Geographic Latitude Longitude projection. If a drawing is created when the project pane has the focus, by default it is created using the Orthographic projection. When the objects were pasted into the Orthographic drawing they appeared as they would be shown in that projection.

Situations like the above arise when we selected the objects in the lat-lon drawing, chose Edit - Copy and then clicked on the project pane and chose File - Create - Drawing. At the moment we clicked on the project pane we moved the focus to the project pane and so the new drawing was created in Orthographic projection. What we should have done was choose File - Create - Drawing while the lat-lon drawing was still the active window. In that case, the new drawing would have been created by default to match the drawing that was open and had the focus.

Note that the pasted North and South America objects are correctly seen in default Orthographic projection, which is, after all, a view from space centered on the intersection of the Equator and the Prime Meridian just off the coast of Africa. From that position North and South America appear greatly foreshortened as they are on the edge of the visible part of the Earth. The default Orthographic projection uses a "transparent globe" so that Alaska in the drawing at right is seen through the Earth. See the Projections topics for more information.
We can retrieve the situation by re-projecting the drawing on the right into Geographic Latitude Longitude projection.

See Also

See the Edit - Paste / Paste Append topic for information on the **Paste Objects** dialog, which pops up when copying objects from one drawing and pasting them into another drawing.
Adding Points, Lines and Areas

There are many ways of adding points, lines and areas to drawings in Manifold. This topic explains how to draw these objects free hand using the Tools toolbar for drawings.

To draw new objects begin by clicking on the map layer tab in which the areas are to be created to make it the active layer. New objects are created in the active layer. It is a good idea to create new objects in a new, blank layer so that they may be easily moved up and down in the layer stack. New objects will be created using whatever formatting is specified for that drawing or map layer.

To add points to a drawing:

1. Click the Insert Point button
2. Click anywhere you wish a point to be created. Points will be created in the active layer/drawing.
3. When finished inserting new points, click out the Insert Point button.
4. If desired, use the Formatting Toolbar to change the formatting of the points.

Very handy: When adding points to a drawing we will often use Instant Data to also add values to a data field.

To add lines to a drawing:

1. Click the Insert Line button
2. Click anywhere you wish the line to begin. Lines will be created in the active layer/drawing.
3. Continue clicking to grow the line with each click to that next position.
4. Right click to finish the line. The line will end at the last left-click position.
5. Click anywhere to begin a new line, and repeat steps 4 and 5 to make another line.
6. When finished inserting new lines, click out the Insert Line button.
7. If desired, use the Formatting Toolbar to change the formatting of the lines.

To add areas to a drawing:

1. Click the Insert Area button
2. Click anywhere you wish the area to begin. Areas will be created in the active layer/drawing.
3. Continue clicking to grow the area with each click to that next position. The area will always grow so that the preview outline of the area is a closed figure.
4. Right click to finish the area. The area will end at the last left-click position.
5. Click anywhere to begin a new area, and repeat steps 4 and 5 to make another area.
6. When finished inserting new areas, click out the Insert Area button.
7. If desired, use the Formatting Toolbar to change the formatting of the areas.

Example: Add an Area

Click on the beginning position of the area and then click at the first position where it is to change shape. Keep clicking about the periphery of the area. As we move the mouse the boundary preview figure will "rubber band" to the current mouse position from the starting point. The rubber band is a preview of what the area will look like if the last left click is made at the current mouse position.
At the end of the desired figure, we would click once to mark the last position and then right click the mouse to create the area.

The area will be drawn within the figure.

Creating Complex Areas

We may wish to create areas that contain "holes" or which consist of several non-contiguous pieces ("islands"). Such areas are called branched or complex areas. To create such areas in Manifold, use a CTRL-click between several sequences of boundary line clicks. Suppose in the example above we wished to create a pentagonal area that contained a pentagonal hole:
At the end of clicking our way around the outer periphery, instead of right clicking we would CTRL-click. We would then click once at a new starting point and click to the first position to change direction. Just as before, as we click our way around the periphery of the "hole" the boundary preview line will "rubber band" from the starting position to the current mouse position.

At the last position we would click once to mark the last position and then right click to end the figure.
Manifold knows that after the CTRL click we have drawn a figure within a figure, so it knows this is to be a hole and creates the complex area accordingly. Had we drawn the second figure outside the first figure, it would have been drawn as an “island.” In fact, we can draw complex overlapping figures by CTRL clicking between them to create new areas of great topological complexity. Note: to keep things reasonably obvious it is usually best to create complex areas with holes or islands but not with complex internal overlaps.

CTRL click also works during the creation of lines. In such cases, a group of what appear to be separate lines will all be created as a single line object. This capability should only be used by experts (and sparingly, at that) since it quite easily leads to confusion.

**Insert Line Sequence**

Curved lines in Manifold are really *polylines*: what appears to be a curve is really one line object that is a sequence of straight portions between coordinates drawn in a “connect the coordinates” fashion.

At times we wish to create a sequence of straight line segments where each straight segment is a separate line object. **Insert Line Sequence** does this. It works just like **Insert Line**, except that it creates a new line object between each click.

Creating a line using **Insert Line** with a few clicks and a right click will result in a single line object.

Using **Insert Line Sequence** with the same clicks and a right click will result in three line objects (the middle of which has been moved to a new layer and colored orange to show it is a different object from the other two straight segments).

**Creating Straight Lines between Points**

Either **Insert Line** or **Insert Line Sequence** may be used to rapidly create a series of straight lines between points.

Suppose we need to create a series of straight lines between the points above. Begin by setting **Snap to Points** so that the mouse cursor will snap to the nearest point.
Make each line by clicking on the first point, clicking on the second point and then right clicking. The resultant "Click, click, right-click" pattern is very fast.

**Status Bar Reports**

When moving the mouse in creating a vector shape the status bar reports the length and angular bearing of the current segment as well as the location of the mouse cursor. Pressing **CTRL** will convert the report to use of alternate units (English or Metric) and will also enable use of ellipsoidal trigonometric calculations for the report.

For example, when creating a line as seen above...

![Status Bar Report Example](image)

...the status bar will report the mouse cursor location followed by **L:** length and **A:** angular bearing. The mouse cursor is located 75.1393 meters from the beginning of the segment at an angle of 43.88 degrees. The mouse cursor is also located at the latitude and longitude indicated.

**Notes**

Using Snap is a great way to guarantee positional accuracy when creating objects free hand.

By default, the **Tools - Options** parameter **Automatically Set Insert Mode** is checked ON. Automatically Set Insert Mode will automatically switch shape modes whenever an insert shape tool button is pressed to that the shape mode that is typically used with that tool. For example, when Automatically Set Insert Mode is ON, pressing **Insert Line** will automatically push in the **Create Lines** mode button and push out the **Create Areas** and **Create Points** buttons. This default choice may be overridden by choosing whatever combination of shape modes is desired after choosing the tool.

In another example, when Automatically Set Insert Mode is checked ON the **Insert Areas** and **Insert Box**, **Circle**, and **Ellipse** commands will create areas only. To create lines and points as well simply push IN the **Create Lines** or **Create Points** buttons after choosing these commands.

**See Also**

- See the Tools toolbar topic for more information on shape modes. See the Freehand Tracing topic for an example use of the **Insert Area** command.
- See the Instant Data topic to learn how to quickly add data attributes when creating an object.
- See the Snap To topic for information on using **Snap** to snap the cursor to exact locations.
- See the Add Points with Instant Data example topic for an example of adding points, using **Instant Data** and moving points by editing intrinsic fields.
• See the Autocomplete with ALT for an important shortcut in the **Insert Area** and **Insert Line** commands when creating areas or lines that fit exactly into existing areas or lines.

• See the Adding Shapes topic to learn to use **shape mode** buttons to automatically create objects as any combination of areas, lines and points.

• See the Dialog Mode and Visual Tools topic for a parameter or value oriented way of creating objects.
Adding Shapes

The commands in the Tools toolbar are used to insert new points, lines and areas into a drawing. Objects may be created free hand or pre-built shapes may be inserted.

- To create areas, lines or points, simply choose the appropriate Insert command and draw the desired shape using left mouse clicks. A right mouse click indicates you are finished and the shape is to be created.
- To create a pre-built shape, choose the shape command desired and then click and drag open a mouse box showing the placement and size of the desired shape.

Shape mode buttons allow automatic creation of the shapes using areas, lines, points or any combination of the three.

See the Adding Points, Lines and Areas topic for additional details and examples. See the Instant Data topic to learn how to quickly add data attributes when creating an object.

- **Insert Area**
  Draw areas within the mouse cursor region defined by a series of clicks followed by a right click.

- **Insert Freeform Area**
  Draw areas within the mouse cursor region defined by clicking and dragging, followed by a right click.

- **Insert Freeform Line**
  Draw lines as defined by clicking and dragging, followed by a right click.

- **Insert Line**
  Draw lines as defined by a series of clicks followed by a right click.

- **Insert Line Sequence**
  Draw lines as a sequence of separate lines.

- **Insert Point**
  Draw points - click to create a point.

- **Insert Box**
  Draw a rectangular box shape with a click and a drag.

- **Insert Box on Center**
  Draw a rectangular box shape centered on initial mouse click.

- **Insert Circle**
  Draw a circle with a click and a drag.

- **Insert Circle on Center**
  Draw a circle centered on initial mouse click.

- **Insert Ellipse**
  Draw an ellipse with a click and a drag.

- **Insert Ellipse on Center**
  Draw an ellipse centered on initial mouse click.

- **Insert Geographic Circle**
  Draw a shape centered on the initial mouse click that forms a circle on the Earth. That is, each location on the shape will be the same distance from the center point. May appear to be a non-circular shape depending on the projection of the drawing.

- **Create Areas**
  Create areas when drawing shapes.

- **Create Lines**
  Create lines when drawing shapes.

- **Create Points**
  Create points when drawing shapes.
Shape Modes

The **shape mode** buttons specify what types of objects should be created when a shape is drawn. Push in the mode button for the type of object desired. For example, if both the **Area** and the **Point** buttons are pushed in, then any circles or other shapes drawn will be created using both area objects as well as points at the coordinates defining the shape. This is an extremely useful effect for simultaneously creating area and line objects for complex shapes.

The following examples create the same star shape using **Insert Area** but with different combinations of shape mode buttons:

- **Areas**
- **Lines**
- **Points**
- **Lines and Points**
- **Areas and Lines**
- **Areas and Points**
- **Areas, Lines and Points**
Automatic Shape Modes

By default, the Tools - Options parameter Automatically Set Insert Mode is checked ON. Automatically Set Insert Mode will automatically switch shape modes whenever an insert shape tool button is pressed to that the shape mode that is typically used with that tool. For example, when Automatically Set Insert Mode is ON, pressing Insert Line will automatically push in the Create Lines mode button and push out the Create Areas and Create Points buttons. This default choice may be overridden by choosing whatever combination of shape modes is desired after choosing the tool.

In another example, when Automatically Set Insert Mode the Insert Areas and Insert Box, Circle, and Ellipse commands will create areas only. To create lines and points as well simply push in the Create Lines or Create Points buttons after choosing these commands.

Shapes, Geographic Circles and Coordinate Systems

With the exception of the Insert Geographic Circle command, all shapes are drawn using the local coordinate system of the drawing. That means that when we insert a circle using the Insert Circle tool a circle is drawn on the XY plane of whatever coordinate system is in use. The result is exactly the same as if we drew a circle on a paper map showing the same region with the same projection.

In many cases, if we are viewing a reasonably small region and the projection in use preserves lengths and other geographic characteristics (and projections are normally chosen so as to preserve such characteristics), drawing a circle in that region will be very close to the shape of a true circle drawn on the Earth's surface at that region. However, because it is not possible to represent the three dimensional, ellipsoidal shape of the Earth in a two dimensional plane (such as a paper map or a computer monitor) without distortion if we draw a circle on a projected map showing a large region of the Earth the figure we draw will not form an actual circle on the surface of the Earth.

To draw a circle that is guaranteed to be a circle no matter what projection is in use we should use the Insert Geographic Circle command. This command allows us to draw a circle of given Earth radius centered on a specified location. The actual shape created in the projected view may be considerably different than a circle depending upon the distortion introduced by the projection, but the object will be a true circle on the Earth's surface.

Example

Consider a map of the Earth drawn in Sinusoidal projection.

Regions far from the center of the map are significantly distorted. In the above illustration we've added a point on the Southwest coast of Australia.

Using Snap to Point we use the Insert Geographic Circle tool to create a circle centered on that point.
The **Geographic Circle** dialog will open already loaded with the coordinates of the central point in the local coordinate system. We choose a radius of 1000 miles and set the number of divisions to 64 to create a nice, smooth object. Press **OK**.

The result is that a very non-circular object appears in the drawing. However, the object has been calculated so that if we looked at it from directly overhead, that is, with none of the distortion introduced by the Sinusoidal projection, it would be a circle.

We can see that is the case by re-projecting the map into Orthographic projection centered on the point that was the center for the geographic circle.

(We can find the latitude / longitude of the point by snapping the cursor to the point and reading the lat/lon value on the status bar. For a cleaner map image in the illustration above, we’ve also re-projected the drawings used and clipped coordinates.)

The Orthographic projection shows us what we would see on the Earth's surface if we were hovering in space over the central point. Clearly, the object created is indeed a circle on the Earth.
To see how the Insert Geographic Circle command differs from the Insert Circle command, we can use the Insert Circle command to create a freehand circle above and to the right of the geographic circle.

Note that in the projected view, the circle we have created appears to be a circle. However, it is just as if we had drawn a circle in that location on a paper map that used the Orthographic projection. The region of the Earth that is covered by a circular figure drawn on such a projection map is not a geographic circle.

If we re-project the map view to use an Orthographic projection centered upon the approximate center of the new "circle" we drew, we can see that the object we drew does not cover a circular region of the Earth.

Tech Tip

To avoid confusion, it is best to use Insert Geographic Circle with projected drawings and maps. If used with latitude / longitude coordinate systems the results can be confusing because degrees are different sized in different parts of the Earth.

See Also

- See the Snap To topic for information on using Snap to snap the cursor to exact locations.
• See the Autocomplete with ALT for an important shortcut in the **Insert Area** and **Insert Line** commands when creating areas or lines that fit exactly into existing areas or lines.
• See the Dialog Mode and Visual Tools topic for a parameter or value oriented way of creating objects.
Instant Data

**Instant Data** make repetitive data entry a snap in drawing windows or in drawing layers in map windows. This is a mode command. It is either checked or not checked in the **Edit** menu. When checked, all subsequent object creation commands, like **Insert Point**, will call up the **Object Fields** dialog, ready for action.

When an object is created, the **Object Fields** dialog in **Instant Data** mode allows us to enter values rapidly into desired fields. The dialog will be pre-loaded with most recently used data and the cursor will already be in editing mode within the most recently edited field. This makes it easy to rapidly enter repetitive values when adding lots of points or other objects to a drawing. For example, if we are using the GPS Console to automatically fetch location data when adding information on, say, fire hydrants or other assets in a town, **Instant Data** makes it easy to rapidly add other annotations.

![Object Fields dialog](image)

**To use Instant Data**

1. Open a drawing window or click on a drawing layer in a map to make it active.
2. Click **Instant Data** in the **Edit** menu to check it ON.
3. Choose a tool to insert the objects you want, for example, **Insert Point**.
4. Click the location where you wish to insert a point. The **Object Fields** dialog will open.
5. Double-click into the value box for which you want to edit data and enter the data.
6. Double-click into any other value box for which you want to edit data and enter the data.
7. When finished adding data for all fields, press **Enter** or press **OK**.
8. Continue adding objects. Each time the **Instant Data** dialog will appear preloaded with the cursor ready to edit within the last value added. Change the values as desired. If for some objects you wish to add a value in a different field, double click into the value for that field and edit it.
9. When finished using **Instant Data**, click it off in the **Edit** menu.

Use **Instant Data** when you must create many objects and wish to add a value for each. For example, suppose we need to add many points to a map and we wish to enter a name for each in a field called **Name**. Check **Edit - Instant Data** and then we can use **Insert Point** to click where we want each point. Enter the name, hit **Enter** and click on the next point. **Instant Data** thus allows a very rapid repetitive data-entry rhythm: click a point - enter a value, click a point - enter a value, and so on.

**Important:** Please note that because **Instant Data** is used to load values into a database field for newly-created objects the **Object Fields** dialog only appears when there are fields defined in the drawing’s table. If we create a new, blank drawing the **Object Fields** dialog will not be available until we add some fields to the drawing’s table.

**Instant Data** remembers the last value used for each drawing. If we turn on **Instant Data** and then add points to different drawing (say, by clicking into different drawing windows or by changing the active layer in a map), the **Object Fields** dialog will be loaded with the last field used and last value used for each drawing.
Using Instant Data when Several Objects are Created

Manifold commands can be used to create many objects at once. For example, if the Insert Areas tool is used with Create Areas, Create Lines, and Create Points modes on the area with a line object at the boundary and point objects at the coordinates that define the area.

When several objects at once are created, the instant data dialog tries to identify the “main” object for which a data field value is to be assigned, using the order of precedence: areas, lines and finally points. For example, when creating an area with a boundary line and points at the defining coordinates the area will be taken as the "main" object and the instant data dialog will set a data field value for the area.

If no object can be obviously identified as the "main" object the instant data dialog will not launch even if Edit - Instant Data is checked. For example, we can use the Insert Box tool with Insert Points mode set ON to create four points at the corners of a box. There is nothing to distinguish any of the four points as being more "main" than any of the other points, so the instant data dialog will not launch in this case.

Example

Suppose we are creating a map that shows the location of fire hydrants in our town. We have a drawing that overlays a high-resolution aerial photograph in a map. Using Insert Point we will click on each location that has a hydrant to create a point at that location. Our drawing's table has fields called Asset and ID Number we can use to identify each point. We would like each point to be created with Hydrant in the Asset field and a varying identification number in the ID Number field.

This is easy. We set Instant Data on and then click to insert the first point.

Since this is the first point to be created in the drawing, the Object Fields dialog launches with no values, but with the cursor ready to enter a value for the Asset field. We enter “Hydrant” and then double-click into the Value field next to the ID Number field.
We now enter the desired value for the **ID Number** field and then press **Enter**. We've just created a point with the desired values.

We can now click into the drawing once more to create another point at the next hydrant location using **Insert Point**.

Once again the **Object Fields** dialog launches, but this time the fields are pre-loaded with default values (the last used) and the cursor is already positioned in editing mode within the **Value** box for the **ID Number** field. We can rapidly change the ID number for this hydrant and press **Enter** to create this next point.

Obviously, if there are many more fields than just one or two that must be filled in with default values when manually adding objects to drawings, **Instant Data** can save a lot of time.

**Note:** When finished using **Instant Data**, remember to take a moment to uncheck **Edit - Instant Data**

**See Also**

See the Add Points with Instant Data example topic for an example of adding points, using **Instant Data** and moving points by editing intrinsic fields.
Editing Objects

Objects in drawings (areas, lines or points) may be edited in several ways:

- Use of Interactive Editing to interactively move or reshape objects by interactive dragging of coordinate locations as discussed below.
- Use of the Edit Toolbar during interactive editing. The edit toolbar provides powerful commands that should be mastered for rapid interactive editing. If not already displayed, the edit toolbar will appear when an object is selected for editing with a CTRL-ALT clicking the object.
- Use the Object Coordinates dialog to manually edit the coordinate values that define the object.
- Edit the coordinates of objects (usually points) by changing their values using intrinsic fields in a table.
- Changing objects using the Transform toolbar or commands such as the Dissolve command.
- See the Dialog Mode and Visual Tools topic for information on using dialog mode when creating new objects.

Objects are Defined by their Coordinates

Objects in drawings are defined by the coordinates necessary to draw them. Coordinates are simply the numbers for each X and Y position that defines the object.

Consider a drawing that contains an area, a line and three points.

The objects are defined by the coordinate numbers that are used to draw them in a "connect the dots" fashion. The area, for example, is nothing more than a list of five coordinates:

3 7
2 9
3 10
5 9
5 8
Through various clever means Manifold knows how to interpret the above list of coordinates as an area and how to draw the area knowing correctly what is the inside and what is the outside of the area.

We can see the coordinate numbers that make up an object at any time using the Object Coordinates dialog. To use this dialog, right click onto the object in a drawing window and choose Coordinates from the context menu. This launches the Object Coordinates dialog that lists all of the coordinates that define that object.

Although it is not the user-friendliest way of editing objects, if desired we can change the shape of an object by editing the coordinate numbers that define it. To edit an object by changing the coordinates manually, double click into any coordinate value to change it. See the Object Coordinates topic for additional information on this dialog.

Points are easier to edit via coordinates than lines or areas because points are defined by a single coordinate pair. It is conceptually simpler (as well as less keyboarding) to change the location of a point by changing the latitude and longitude of the point than it is to edit a long series of coordinates for a line or an area. We can move points by changing the latitude and longitude of their location as shown in their intrinsic fields in a table. We can move lines and areas by changing their latitude and longitude intrinsic field values in a table; however, doing so moves the entire line or area by moving its centroid and does not reshape the object. See Editing Intrinsic Fields in Tables.

**Interactive Editing**

**Smart Mouse** touch selection is used to select objects for editing. An object selected with smart mouse selection will appear with edit handles at each coordinate position that can be used to move or reshape the object. An object chosen for editing is also called the **primary selected object**.

To select an object as the primary selected object for editing, exit all command modes so that the mouse cursor is in the default arrow form and then CTRL-ALT click on the object to be edited. Edit handles will appear on the selected object at the location of each coordinate that defines the object. We can then move these handles about to change the shape of the object. This is a lot easier than editing coordinate numbers manually via the Object Coordinates dialog.

If we change our minds and we don’t want an object to be the primary selected object for editing, we can deselect it by CTRL clicking on it. A CTRL click on an object will toggle the selection status of that object (selects it or deselects it) without changing the selection status of any other object.

**Keyboard Modifiers with Smart Mouse Touch Selection**

Because a selection may already exist in the drawing when we wish to select an object for editing, smart mouse selection uses keyboard modifiers to allow a richer set of commands.

- **Click** Equivalent to Select Touch in Replace mode. Click on an object to select it. All other objects are deselected.
  
  Clicking into an empty part of the drawing deselects all objects (with confirmation dialog).

- **CTRL-ALT-Click** Select an object for editing. Click on an object to select it as the primary selected object.

- **CTRL** Invert the selection state of the object without changing the selection state of any other object. Equivalent to Invert mode.

- **SHIFT** Select the object if it is not yet selected. Does not change the selection state of other objects. Equivalent to Add mode.

**Editing Commands**

The primary selected object may be moved or changed in shape by clicking and dragging its editing handles:

- **CTRL-click and drag** CTRL-click an editing handle and drag to move the entire object.
SHIFT   Hold the SHIFT key down while clicking an editing handle and dragging to move the entire current selection, which includes the primary selected object as well as all other selected objects in the active drawing or drawing layer.

ALT-SHIFT The ALT key extends the action of the SHIFT click and drag to all visible layers. Moves all selected objects in all visible layers.

Click and drag Click an editing handle and drag to move just that handle and so reshape the object.

CTRL-SHIFT Holding down the SHIFT key while using CTRL-click to move an ending handle will move only that particular branch in multi-branched objects.

Right-click Right-click an editing handle to call up a context menu of additional editing choices. These will allow deletion of coordinates, for example, via the Coordinate sub-menu.

Right-click a segment between editing handles to call up a context menu allowing choices such as adding a new editing handle in that segment.

Editing works by first clicking on the object to make it the primary selected object. Edit handles will appear at the coordinates that define the object. One may then move the entire object or move the edit handles to change its shape. Edit handles may be deleted or added via the commands in the right click context menu. In the case of branched objects (which include islands or holes in area objects), one may delete branches or move them about.

Note that ALT-SHIFT editing will apply to all objects in visible layers, whether or not those objects are visible at that moment due to Zoom range or Layer Opacity settings. It is possible to have a layer be enabled for display (that is, the layer tab was not clicked off) but the objects within that layer not be visible because the layer's Opacity was set to zero percent. Likewise, a layer's objects might not be visible because of a Zoom range setting. In both cases, the ALT-SHIFT editing will still apply to any such invisible objects.

Editing Example

We begin all smart mouse selection and editing by clicking out any mouse command mode. This restores the mouse to the default arrow cursor.

To edit an object we CTRL-ALT-click on it. In this example we CTRL-ALT-click on a triangular area.

The triangular area becomes selected and shows edit handles at the coordinate positions that define the area. This is a simple area that is defined by only three coordinates. A more complex area would show an edit handle at each coordinate that defines it. Larger areas in geographic drawings are often so complex and will have so many edit handles appear that they will blend together in what might appear, at first glance, to be a very wide border.
To move the object, we **CTRL**-click and keep pressing on any edit handle. Note that the entire outline of the area changes color to show the entire object can be moved.

Without releasing the mouse button, we drag the object to whatever new place is desired. A preview outline shows where the drawing will go.

When we release the mouse button at the new position the object moves to that location.

We can edit the shape of the object by clicking and dragging the edit handles. If we click on the edit handle as shown above the area borders adjacent to the edit handle change color to show that these borders will be altered by moving the edit handle.

Dragging the mouse button shows a preview of how the shape will be altered when this edit handle is moved.

Releasing the mouse button causes the area to change shape to the new position of the edit handle.
To deselect the object, we can either **CTRL**-click on the object to toggle the selection state off, or we can simply click anywhere in "empty" space in the drawing to deselect all objects.

The result is a reshaped object in a new location.

**Selecting Boundary Lines**

Beginners at GIS often confuse the idea of an area object with the idea of a region enclosed by a line object. Please read the Drawings topic to clear up this confusion if you have not already done so.

When using smart mouse selection and editing we must click on the object that is to be edited. Suppose we have a line that forms a closed figure. If we wanted to select that line for editing we must click on the line itself and not simply click within the enclosed region as if it were an area object.

Clicking within the enclosed region has no effect except deselecting all items (if we confirm the desired deselection in the dialog that pops up).

To select the line, we must click on the line itself.
This results in the selection of the line object for editing.

Errors of this nature may be avoided by working with the **Background** off in the layers pane so it is more clear that there are no objects inside lines that happen to form enclosed regions.

**Selecting for Editing**

We may wish to edit an object within a selection without changing the selection. We can choose objects for editing using smart mouse selection with a **CTRL-ALT** click without changing the selection state of other objects. Suppose, for example, we have made a selection in a drawing showing Denmark.

**CTRL-ALT** click on an object to select it for editing as the primary selected object. Here we **CTRL-ALT** click on the island of Langeland to select it for editing.

When it is selected for editing, edit handles will appear at all coordinates that define the object. Note that the selection state of all other objects has not been changed.

**CTRL-ALT** click on a different object to select it for editing. Here we **CTRL-ALT** click on Falster island.
CTRL-ALT clicking on Falster selected it for editing while leaving Langeland selected along with the other objects.

**Context Menu Editing Commands**

Right clicking on an object selected for editing calls a context menu that contains advanced editing commands. When right clicking on an editing handle, the context menu will also include editing commands for that coordinate.

**General** (Right clicking onto object)

- **Duplicate**
  Duplicate the object. It is wise to immediately move the new object while it is selected. This avoids creating coincident objects that might later cause confusion.

- **Orthogonalize**
  Move coordinates to orthogonal grid positions. Used to trim insignificant digits.

- **Segmentize**
  Add redundant coordinates to increase the number of coordinates used to define the object. See the Segmentization topic for discussion.

- **Simplify**
  Remove coordinates that define the object to simplify the shape of the object.

- **Center**
  Pan the view to center this object in the window.

- **Zoom**
  Zoom to fit this object.

**Coordinates**

Launches the Object Coordinates dialog to allow direct editing of the coordinates that comprise the object. Right clicking onto an edit handle and choosing **Coordinates** will launch the Object Coordinates dialog with that particular edit handle coordinate highlighted.

**Fields**

Launches the Object Fields dialog that shows any table fields associated with this object.

**Coordinate** (Right clicking onto an edit handle)

- **Delete**
  Delete this coordinate. It is not possible to delete the last coordinate necessary to define a point (one coordinate) a line (two coordinates) or an area (three coordinates).

- **Delete / Split Branch**
  Delete this coordinate splitting the branch it is in into two branches. Only available when editing a line object. It is not possible to split the branch by deleting its first or last coordinate, the coordinate after the first coordinate or the coordinate preceding the last coordinate.

- **Duplicate**
  Duplicate this coordinate. It is wise to immediately move the new coordinate/edit handle to avoid confusion that might be caused by coincident (redundant) coordinates.

**Coordinate** (Right clicking onto a segment between edit handles)

- **Add**
  Add a new edit handle (coordinate) at the location on the
segment that was right clicked. Such a click will not likely be exactly on the line segment, as a click reasonably near a selected object will be interpreted as involving that line even if the click was not exactly on the line segment.

**Add on Segment**  Add a new coordinate to the object exactly on the line segment nearest to the click.

**Add Mid-Segment**  Add a new coordinate to the object exactly in the middle of the line segment nearest to the click.

---

**Branch**  (Right clicking onto a branch or edit handle)

**Delete**  Delete this branch. It is not possible to delete the last branch of an object.

**Duplicate**  Duplicate this branch. It is wise to immediately move the new branch to avoid creating coincident branches that might later cause confusion.

**Split**  Enabled only when right clicking onto a coordinate of a line object. Splits the line into two branches at this edit handle. This is an expert-level command that is not the same as splitting the line into two line objects at this position.

---

**Segment**  (Right clicking onto a segment between edit handles)

**Delete**  Delete this segment. It is not possible to delete one of the last two segments of a line branch or one of the last three segments of an area branch.

**Delete / Split Branch**  Delete this segment splitting the branch it is in into two branches. Only available when editing a line object. It is not possible to split the branch by deleting its first or last segment.

---

The above context menu commands should be used in combination with the Edit Toolbar interactive commands.

**Changing the Shape of Objects**

We can change the shape of objects by moving existing editing handles, by deleting editing handles or by adding new editing handles, which may then be moved to new positions.

Suppose we would like to change the shape of a selected area. We would like to add an editing handle, so we right click at the location in the segment where we would like to add the handle.
We choose **Coordinate - Add** from the context menu to add a new coordinate and editing handle to the clicked segment at the place clicked.

We can now click and drag to move this new editing handle.

For example we can drag it out to the right. A preview line shows the new shape of the area as we drag.

When we release the mouse button at the end of the drag the area changes shape to fit the new location of the edit handle.

To delete a coordinate and associated edit handle we right click on the handle and choose **Coordinate - Delete**.
This deletes the coordinate at that position. The area will immediately change shape to the shape defined by the remaining coordinates.

**Editing Branched Objects**

This is an advanced topic. A **branched** object is one that includes islands or holes as one object. What appear to be separate objects are really the same object but implemented using a complex internal definition comprised of several sequences of defining coordinates all concatenated into one object. Each sequence of coordinates is called a branch.

The usual case is a state or some other geographic region that includes islands. Instead of each island being a separate object, they are all formed into a single area object.

When selecting objects for editing it is easy to see if what appear to be separate objects are in fact a single, branched object. CTRL-ALT clicking on an object in smart mouse selection will cause all the branches to appear with edit handles if they are all different branches of the same object.

In this example, we've CTRL-ALT clicked on one of the three islands and all have appeared with edit handles. This is a certain indication they are all just branches of a single area object.

Right clicking on one of the branches and choosing **Branch - Delete** from the context menu will delete the branch.
The other two branches will be left unchanged.

The example above shows use an area where multiple branches are used to create islands. Multiple branches are also used to create "holes" within areas. Clicking and dragging an edit handle will move the entire object, including all branches. This default behavior assures that all holes in areas that use branches to define internal voids will be correctly moved.

We can use a CTRL-SHIFT click on an editing handle and drag to move individual branches, whether they are used to represent islands or holes.

Suppose we have a rectangular area with an internal, triangular hole. The hole is a separate branch from the outer border of the area.

We can CTRL-ALT click on the area to make it the primary selected item so that edit handles appear. We can then CTRL-SHIFT click and drag on an edit handle of the internal hole to move just that branch.

While dragging, a preview will show us how the branch is being moved.

When we release the mouse the branch will move to the new position. In this case, we have moved the position of the hole in the area.

Note that we could use the same CTRL-SHIFT-click and drag on an edit handle to move the relative positions of the "islands" in the previous example.

Moving Objects by Changing Intrinsic Fields

Manifold tables can show intrinsic fields that are computed from the object, such as the position of the center of the object. See the Editing Intrinsic Fields in Tables topic for a discussion of how we can move objects by editing their locations as they appear in intrinsic fields in a table.
Keyboard Zoom Commands during Selection and Editing

When making selections or editing objects (such as inserting areas using the Autocomplete feature) we will often want to zoom in or out in the middle of a selection or editing command. Use the + and - keys on the keyboard to do so. Usually the + and - keys in the numeric keypad are used.

+  "Plus" key: Zoom in at the current tool position.
-  "Minus" key: Zoom out at the current tool position.

The current tool position is the position of the mouse at the time the + or - key is pressed. When editing, using these keys allows us to easily zoom into a desired location, make a few precise clicks and then zoom back out and proceed at the usual scale.

Notes

See the Selection in Drawings topic for additional information on smart mouse selection.

This topic discusses interactive editing of objects using the mouse. However, as a reminder for the reader we should note the following when using non-interactive editing methods such as transform operators or menu commands.

Many editing / transformation operations in drawings (see the Transform Toolbar - Drawings and Drawings - Dissolve topics, for example) will create new objects from existing objects. When new objects are created from existing ones the system must know how to transfer data attribute fields from the existing objects into the new objects. The Transfer Rules dialog in tables sets forth the rules to be used for each particular column.

See Also

Drawings - Object Coordinates
View - Structure
Edit Toolbar
**Editing with Snap**

The Snap To cursor modes are very important when editing objects. Before proceeding, please review the Snap To and Editing Objects topics.

Beginners should practice editing using the interactive commands described in Editing Objects until they have perfect fluency without any Snap modes set. Snap modes can be confusing to beginners because the action of the command takes place at the snap cursor and not at the mouse. It therefore is imperative that the new user learns basic editing motions before beginning use of Snap.

When editing with Snap we will often wish to use a mix of mouse commands so that sometimes Snap is on and sometimes Snap is off. At times we will want to toggle Snap on and off in the middle of a mouse motion.

To temporarily turn off Snap we use the space bar. Pressing the space bar will turn Snap off even in the middle of a mouse motion. Press the space bar again to turn Snap back on. To edit effectively with Snap we must therefore practice so that use of the space bar becomes second nature. Beginners may be confused until they have enough experience to use the space bar correctly without having to think about it. This is a simple neuromuscular skill that is best learned through repetition.

**Note:** The snap toolbar is turned off by default. To turn it on, open a component that uses the snap toolbar (such as a drawing), choose Tools - Customize and check the box for the snap toolbar.

**Example: Reshaping an Area to match Another Area**

We first consider an example using Snap that does not use the space bar. Suppose we have two rectangular areas near each other in a drawing:

![Example 1](image1)

We would like to reshape the area on the left so that along its right edge it perfectly matches the area on the right:

![Example 2](image2)

We can accomplish this with simple interactive editing, and by setting Snap to Areas. This snap mode moves the cursor only between coordinates that define areas.
Using smart mouse editing we **CTRL-ALT** click on the rectangle on the left to select it as the primary selected object. (See Smart Mouse Selection for notes on "smart mouse" moves.) Edit handles will appear at all coordinates that define the area. Note that as we bring the mouse near one of the coordinates that define either area the snap cursor will jump to that coordinate.

We’ll move the lower corner first. To move it, we click the edit handle and drag down and to the right. It’s easy to click exactly on the edit handle because the **Snap to Areas** mode jumps the cursor directly onto edit handles.

As we continue to drag down near the corner of the second area, the secondary snap cursor (the box) jumps to that coordinate.

When we release the mouse button, the area will be reshaped to adjust to the new position of the edit handle.
We repeat the process with the upper edit handle by clicking on it and dragging to the right.

As the mouse nears the other area the secondary snap cursor will jump to the coordinate of the corner.

When we release the mouse button the area adjusts to the new position of the edit handle.

If we deselect the area we can see we have achieved our objective.

**Note:** Clicking and dragging an edit handle moves just that handle to reshape the object. CTRL-clicking and dragging an edit handle will move the entire object.

**Example: Moving Points with Snap**

In the example above we did not use the space bar to toggle Snap off temporarily. We did not need to do so because we were editing areas and it was convenient that Snap to Areas jumped the cursor between coordinates (and thus edit handles) in either of the areas involved. In many situations we will need to temporarily turn Snap off during an editing command.
Suppose we have a drawing with some points near an area. Perhaps the area represents a coastline. We would like to move the middle point onto the coastline using **Snap** to assure that it is located precisely on the coastline.

If we CTRL-ALT click on the area object to select it for editing using smart mouse selection, we can see edit handles appear at the coordinates that define the area. We can then **CTRL** click the area to turn off selection for editing.

If we set **Snap to Areas** the snap cursor will jump between these coordinates.

With **Snap to Areas** set, no matter where we move the mouse the snap cursor will move only between coordinates in the area.

To temporarily turn **Snap** off, we press the space bar. For example, to select the middle point for editing using smart mouse selection we can press the space bar and then click the point. The space bar temporarily turns off **Snap** so the mouse can click on the point.
To move the point, after pressing the space bar we can click onto the point and drag it towards the desired location. The magenta snap cursor in the illustration shows the snap to position before we pressed the space bar. The secondary box cursor shows the destination of the edit move.

With snap temporarily turned off, the secondary box cursor moves with the mouse.

At any time we can press the space bar again to turn snap back on. If we do so, the box cursor will jump between only allowed Snap to Areas coordinates. If desired, while dragging we can press the space bar several times to turn snap off and on again and so cause the box cursor to jump back and forth between the mouse cursor and allowed Snap locations. As long as we do not release the mouse button we can go back and forth as many times as we like.

To move the point to the location denoted by the original snap cursor, after pressing the space bar a second time to turn snap back on we can drag near that location. The box cursor will Snap to that location.
When we release the mouse button to end the dragging motion the point moves to the new location.

If we deselect it we can see it is now in a new place.

**Snap Tolerance**

The **Snap Tolerance** parameter may be optionally specified in the Tool Properties pane to control how close the mouse must be to a snap item before a snap mode takes effect.

When a snap mode is on, the **Snap Tolerance** setting gives the distance in screen pixels or in physical units within which the cursor must be to a given snap item (such as a line when snapping to lines) before the cursor snaps to that item.

When the cursor is outside the snap tolerance range, snaps are not effective. Setting a small snap tolerance is therefore a good way of retaining the benefits of snap modes without having the snap cursor jump about in a disconcerting way if the mouse is nowhere near a snappable item.

**Layer Snap Restrictions**

Using Layer Restrictions a map layer can be made non-snappable. When a layer is so restricted, any snap modes will ignore items in that layer.

**See Also**

**Edit - Snap To**
Autocomplete with ALT

When editing maps we often would like to create a new area that fits exactly into an existing area. Manifold provides an autocomplete capability within the Insert Area and Insert Line tool to make this easy. When Snap to mode and an Insert Area or Insert Line command is operational, pressing the ALT key will cause Manifold to autocomplete the area or line between the last clicked coordinate and the current snapped cursor position.

To create an area using autocomplete to follow an existing area:

1. Engage Snap to Areas.
2. Choose the Insert Area command.
3. Click on the location where the area is to begin.
4. Holding the ALT key down, click on the location where the area is to end.
5. Right-click to create the area.

Autocompletion using the ALT key requires a small amount of manual dexterity that comes with practise. Practise creating areas in this way until the motion is automatic.

Example

We would like to create a new area in the bay shown above. The new area must match exactly the area surrounding it with no overlaps and no gaps.

We begin by engaging Snap to Areas. The cursor will now move only between the nearest coordinates that define an area.

Choose the Insert Area command. To create an area with this command we left-click onto the coordinates we want to define the area and then we right-click to create the area.

We move the cursor near to the starting coordinate of the new area. The Snap mode jumps the cursor directly to the nearest coordinate of the existing area. We click to mark that coordinate as the starting coordinate for our area.
If desired, we could move the mouse over all of the coordinates in the existing area and click on each. The **Snap** mode will assure that any coordinates we click will be coordinates in the existing area. However, it is very tedious to click on all of the coordinates that define the edge of a complex area without missing a few. Missing any coordinates will result in overlaps or gaps.

If **Snap to Areas** is on, pressing the **ALT** key during the **Insert Area** command instructs Manifold to **autocomplete** the area between the last-clicked point and the current **Snap to** cursor position. All of the coordinates in between will be chosen for the new area. Autocompletion is indicated by a contrasting tracking/marquee line following the edge of the existing area.

As long as we hold the **ALT** key down, the autocompleted edge will be shown in black between the initial point and the current mouse position.

At any time we can lift or hold down the **ALT** key. Lifting it drops the **Insert Area** command back into its default mode, where a straight line edge would be created between the previously clicked coordinate and the current mouse coordinate. Pressing the **ALT** key turns autocompletion back on.
To create the area, while holding the **ALT** key down we move the mouse cursor to the desired ending position of the area and then we click (while still holding the **ALT** key down). We can now release the **ALT** key and right-click to create the area.

The result is an area that is created between the initially clicked point, through all of the coordinates in the edge of the other area and the final clicked point. The new area is shown in blue above.

**Shortest Edge and Longest Edge**

The **ALT** key generates an autocompletion boundary following the shortest path from the initially clicked point to the current mouse position. Pressing the **SHIFT-ALT** key combination generates an autocompletion path following the longest path.

- **ALT** Autocomplete boundary using the shortest edge path.
- **SHIFT-ALT** Autocomplete boundary using the longest edge path. Same as **ALT** when following lines with autocomplete.

Using **ALT** autocompletes the area boundary using the shortest edge. Using **SHIFT-ALT** autocompletes using the longest edge.
With **ALT** the resulting area is a small bay. With **SHIFT-ALT** the resulting area is a large object that encompasses (is a union of) the original object plus the bay.

**Autocompletion across multiple objects**

The autocompleted boundary must be all in the same object. One cannot click to start a new area in one object and then try to autocomplete it in a different object. To create areas using more than one other area as the defining boundary, we must use autocompletion in two steps.

To create an area using the two existing areas in the illustration above we begin by clicking on the starting coordinate in the first area (**Snap to Areas** is enabled and the **Insert Areas** command is in action).

Holding down the **ALT** key we move the mouse over the last coordinate in the first area and we **ALT**-click to mark that coordinate.
Next, we move the mouse cursor slightly downward onto the first coordinate that we would like to use from the second area and we click to mark that location. Note that we simply click on this location and we do not ALT-click.

We can now press ALT and the autocompletion will follow the boundary of the second area from the position of the last click, which is indicated with a red arrow in the illustration above. (Manifold does not draw red arrows as seen above. The red arrow was added to the screen shot.) We can then ALT-click to mark the final coordinate followed by a right-click to create the area.

When using autocompletion across multiple areas it is tempting to work at a high zoom level and to use autoscroll to follow the edge of the areas. When areas are complex, working at a high enough zoom level to be sure to mark the "last" and "next" coordinates of adjacent areas at boundaries between two areas results in a lot of slow autoscrolling.

There are more efficient ways of managing zoom. One alternative is to use the + and - keyboard keys to zoom in and out even in the middle of the Insert Areas command. Pressing + on the keyboard will cause a zoom in. Using the + and - keys we can zoom in where greater precision is required and then zoom out where less precision is required. Using the + and - keys while simultaneously operating the mouse within an Insert Areas command takes some manual dexterity and practise.

A second approach is to work at a zoom level that is convenient for the overall scene and to make a best effort to mark the last and next coordinates at the boundaries between areas. We can then zoom far into the scene at each boundary between areas and add/delete coordinates to the newly created area and move them as desired (using Snap to Areas) to adjust the fine detail of the new area in that location so there are no overlaps or gaps. See the Editing Objects and Editing with Snap topics.

**Autocompletion with Lines**

We can create lines using autocompletion as well as areas. We can also use lines to guide the autocompletion as well as areas.

For example, suppose we have a line that defines a boundary. We would like to create an area using this line as a guide.

We turn on Snap to Lines and Insert Areas. We click on the starting point of the area.
Holding the **ALT** key down to autocomplete along the line we **ALT**-click on the ending point of the area.

We can then right-click to create the area.

We can also create lines using areas as a guide.

To do so we turn on **Snap to Areas**.

We will create the line with **Insert Line**.

Suppose we want to create a line at the boundary of two areas. We click on the beginning coordinate.
Next, we **ALT**-click at the ending coordinate. The **ALT** key will cause an autocompletion along the edge of the area we are following.

Right-click to create the line. It is shown in the illustration above with a line size of 2 to show the new line better. Note that when creating a line on the common edge of two adjacent areas it doesn’t really matter which area is the "upper" area, since both have the same coordinates in common at their adjacent edge.

### Keyboard Zoom Commands during Selection and Editing

When making selections or editing objects (such as inserting areas using the **Autocomplete** feature) we will often want to zoom in or out in the middle of a selection or editing command. Use the + and - keys on the keyboard to do so. Usually the + and - keys in the numeric keypad are used.

+ "Plus" key: Zoom in at the current tool position.
- "Minus" key: Zoom out at the current tool position.

The current tool position is the position of the mouse at the time the + or - key is pressed. When editing, using these keys allows us to easily zoom into a desired location, make a few precise clicks and then zoom back out and proceed at the usual scale.

### Limitations

Autocompletion has several limitations:

- The autocompleted boundary must be all in the same object. One cannot click to start a new area in one object and then try to autocomplete it in a different object. See the notes above when autocompleting across multiple objects.
- Autocompletion works only within the same branch of a multi-branched object.
- When a starting coordinate occurs at a spot where several objects in the same drawing have a defining coordinate, the autocompletion path follows only the topmost object. If desired, we can move an object to a new drawing to force it to be the "topmost".
Notes

The "A" in ALT is a mnemonic for Autocomplete.

The colors used for both the primary marquee (the selection "box") and the autocompletion trackline are set in Tools - Options by the Marquee color. The autocompletion trackline is the RGB color inverse of the Marquee color. When setting Marquee color it is important to choose colors that result in reasonable contrast when the primary marquee and the trackline are seen against the colors used in the drawing.

See the Snap to topic for information on Snap modes.
Shared Edit

The Edit - Shared Edit setting controls the operation of mouse-based editing commands. When Shared Edit is on, any click and drag of the mouse will affect all objects that have a defining coordinate at the mouse click location. When Shared Edit is off, only that object that has been selected for editing will be affected. When Shared Edit is on, read-only layers, non-editable layers and invisible layers are ignored.

Shared Edit may be turned on and off using CTRL-E as a keyboard shortcut ("E" being a mnemonic for "Edit"), or by clicking the Shared Edit choice in the Edit menu.

When Shared Edit is on, a box will appear around the Shared Edit icon in the Edit menu.

When Shared Edit is off, there will be no box around the Shared Edit icon in the Edit menu.

Example

The easiest way to show the effect of Shared Edit is to turn it on and try a simple example.

Consider a drawing that has four lines. Three of the lines have been drawn using the Snap to Lines snap mode so that their ends are exactly at the end of the fourth line. At the ends of the four lines five points have been placed, also using Snap to Lines snap mode, so that the points are exactly at the ends of the lines.

This is a fairly typical example of lines and points that might occur in a drawing of, say, a system of roads or a network, where the ends of lines are exactly coincident and where points might be placed exactly at the ends of lines. Suppose we would like to edit this drawing to move, say, the position of one of the points and we would like to simultaneously change the position of any incident lines.

That would be tedious to do without Shared Edit capability, but with Shared Edit mode turned on it is very easy to do.
Suppose we would like to move the central point and to move all incident lines with it. We begin by `CTRL-ALT` clicking on the point to select it for editing. An edit handle appears.

We can now click on the point and drag it to a new location.

With **Shared Edit** mode turned on, when the point moves, the coincident coordinates of all incident objects move with it. The four lines with ends exactly under the point will have their ends also moved. What is happening is that the coordinates defining the end of each line are also changed when the point's coordinates are changed by moving the point.

This works regardless of which object has been selected, and it even works if the different objects are in different layers and we are editing them in a map.
Suppose, for example, we *CTRL-ALT* click on the uppermost line to select it for editing. Edit handles appear at the two coordinates that define the line.

We can now click and drag the edit handle on the bottom coordinate to move it.

With *Shared Edit* mode on, when we move the edit handle for the line we change the line, but the point at that location also moves and so do the ends of the other three lines.
Let's consider what would have happened if we did not have Shared Edit mode on. Suppose we CTRL-ALT click the upper line to select it for editing.

Once again, we could click and drag the bottom edit handle to move it.

However, if Shared Edit mode is turned off then moving the bottom edit handle will move only the coordinate for that one object and will not move the point and the ends of the other three lines that were coincident with it.

**Editing Areas with Shared Edit**

Shared Edit mode is very useful for editing areas where areas are adjacent. It's often the case with geographic maps that areas will be drawn adjacent to each other. If we want to edit the boundary of an area to move it slightly we will also need to correspondingly move the boundary of any adjacent areas to avoid creating gaps or overlaps.

Geographic regions such as provinces or states are often represented as areas in geographic maps. The "border" between two states in an error-free geographic map is a place where the boundaries of two different areas coincide. Let us digress for a moment to make this concept clear:
The illustration above shows a map that has two layers, each containing an area object. The areas have been drawn so they are exactly adjacent to each other along a common border. What appears to be a single line, a "border," is in fact simply the place where the coordinates of two different area objects happen to coincide.

We can see this by clicking the layers on/off for display that contain the two areas.

One layer contains an area colored in gray. This area is defined with seven coordinates, five of which define the right edge of the area.

The other layer contains an area colored with a blue pattern. This area is also defined with seven coordinates. The left edge of the blue patterned area is also defined with five coordinate locations.

What gives the appearance of a common "border" is that the five coordinates defining the right edge of the gray area are exactly coincident with the coordinates defining the left edge of the blue area.

This is the usual case in most geographic maps. Unless there is a boundary dispute between two states, normally where the territory of one state ends is where the other state begins, so there should not be any gaps or overlaps at the place where two areas meet.

To move the common boundary line, we want to move the coordinates of both the gray area and the blue area simultaneously. To do this, we turn Shared Edit mode on and we will use regular editing commands to change the shape of one area with the other area's shape being changed automatically.
For example, we can select the area on the right for editing by **CTRL-ALT** clicking it so edit handles appear at each of the area's coordinates and then we can drag edit handles about to change the shape of the area.

To move the central part of the border we can click and drag the leftmost edit handle to the right.

With **Shared Edit** turned on, the result is that the corresponding coordinate for the gray area is also moved to the right. This changes the shape of the gray area as well so that no gaps or overlaps appear. Note that this works even though the two areas are in different layers in the map window in which we are working.

Had we done this same edit with **Shared Edit** turned off, only the coordinate for the area selected for editing would have been moved. The coordinate for the gray area would not have been moved and thus a gap would appear between the two areas.
Dissolve

**Dissolve** is so named because it “dissolves away” or removes the borders between areas that have identical values in data attribute fields. Given a set of areas, it can automatically create new areas that are spatial combinations of areas having the same data attributes. **Dissolve** can also work with lines, to join lines together. It is available for drawings or drawing layers in maps.

Suppose we have a drawing of the US where each state has a region field.

<table>
<thead>
<tr>
<th>Name</th>
<th>Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alabama</td>
<td>East</td>
</tr>
<tr>
<td>Arizona</td>
<td>West</td>
</tr>
<tr>
<td>Arkansas</td>
<td>Central</td>
</tr>
<tr>
<td>California</td>
<td>West</td>
</tr>
<tr>
<td>Colorado</td>
<td>West</td>
</tr>
<tr>
<td>Connecticut</td>
<td>East</td>
</tr>
</tbody>
</table>

The value in the region field can be **East, West** or **Central**.

If we **Dissolve** all objects in the map using the **Region** field we combine all areas of like region value into one area. Thus, all the states with the value **West** in their **Region** field will be unioned into a single area.

**Dissolve Dialog Controls**

The **Drawing - Dissolve** dialog allows us to choose which areas are to be dissolved and what the guiding field should be.
**Dissolve**

Choose [All Objects] or the name of a saved selection.

**Using**
Choose the field to use to guide the dissolve process.

**Ignore Case**
Combine text strings treating any variation in case as the same string.

**Ignore leading and trailing whitespace**
Combine text strings without regard to whitespace characters before or after the string.

**Ignore internal whitespace**
Combine text strings without regard to whitespace characters occurring within the string.

Manifold uses the Transfer Rules specified for the table's fields to combine data attributes when dissolving objects into one object. Before using **Dissolve** it is essential to read and understand the Transfer Rules topic.

Different transfer rules will be used with different types of information. For example, suppose each state has an **Area** field that gives the area of the state in square kilometers (or more likely for the US, in square miles…). We would like the **Area** field for the combined area created by **Dissolve** to be the **Sum** of all the areas of the states that were unioned together. To accomplish this, we would right click onto the **Area** column head in the drawing's table, choose **Transfer Rules** in the context menu and then verify that **Sum** is the **N to 1** transfer rule for that column (which it is, by default).

**Dissolving Lines**

Suppose we have a drawing that shows streets as lines. Each street has a **Name** field that gives the name of the street.

It's often the case that each street will be represented with many line objects where the end of each line is exactly coincident with the next line. In the illustration above we can see such a situation by selecting (with select touch) every other line object that makes up a particular street.

If each of the line segments that makes up a given street has the same name, such as "Fisher Street" we can use **Dissolve** to automatically join all the line objects that make up a particular street into a single line object.
To do so, we run **Dissolve** with the settings above.

![Dissolve interface](image)

The result is that all of the line objects that form a given street will be dissolved into a single line object. We can see this by using touch selection to touch the line anywhere and see the entire line selected at once.

**Performance Tips**

- Don't use high-resolution data when lower resolution will do. For example, if you are mainly interested in showing provinces for thematic display purposes and are not really interested in very detailed shorelines a relatively low resolution data set will work fine. Resample the data with Normalize Topology (for drawings) using a lower Location Precision value to create a lower resolution equivalent that makes sense at the display level of interest. **Dissolve** can work much faster if the objects involved are defined with fewer coordinates.

- Re-project all drawings involved into the same projection. All projection parameters must be the same for this to help and not just the name of the projection in use.

- When importing drawings, don't import data fields that will never be used. Many data fields will slow the system down. If a drawing has been imported with superfluous data fields, edit tables using Design to eliminate the unnecessary fields. When working with commands such as Dissolve, check the Transfer Rules for the tables being used to make sure that no fields are being transferred that are not necessary to transfer.

See the comprehensive list of suggestions in the Performance Tips topic.
Districts

The **Drawing - Districts** command assigns areas in the drawing to different districts by writing a district code to a specified data field. Given a drawing that contains areas the command assigns each area to one of the given number of districts such that each district is reasonably compact and contiguous. The **Districts** command is normally used with drawings showing adjacent areas like a drawing of the countries of Europe, provinces in a given country, counties in a US state and so forth.

If you have acquired the optional Business Tools extension to Manifold System, use of the Districts (Visual) dialog is much preferred to the basic **Districts** dialog.

**Scope**
The set of areas to be assigned to districts. [All Objects] by default, but can be set to the selection or any saved selection.

**Save to**
The column to be used to save the district assignment code. [New Column] by default to create a new column called **District**, or a choice of any existing column.

**Balance**
Field to be used to balance districts. Districts will be created so each district (as much as possible) has the same total value of this field.

**Threshold**
Areas within this distance will be considered adjacent. The default value of 0 means that areas touching each other are considered "near" and that areas with no common points are "far" from each other.

**Districts**
The number of districts to which the areas should be assigned. Choosing 4, for example, will cause each area in the drawing to be assigned to one of four different districts.

**Modify formatting**
Reformat colors of areas so that areas in each district are colored similarly. Colors used are taken from the same palette used in the Color dialog.

Choosing 4 districts will cause the Districts command to create a new column called **District** and then place a value of 1, 2, 3 or 4 into that column for each area. All areas with the value 1 will be grouped together next to each other in one district as will the areas with values 2, 3 and 4 respectively. Redistricting can be accomplished by opening the drawing's table and changing the **District** code for any area that is to be reassigned to different district.

**Note:** A value of 0 is assigned to newly-created areas that do not yet participate in any district as well as to lines and points.

The thematic format created by default will use the **District** column with unique values to specify color. To change the color scheme used, use thematic formatting to change the colors used to format using the **District** column.

**Example**

We will create districts using the **mexico_eg** sample drawing (referred to here as "mexico" for brevity).
Seen in default formatting in a drawing window, the *mexico* drawing consists of areas for provinces, each of which has a variety of columns giving geographic and demographic information.

Choose **Drawing - Districts** and choose **SQKM** as the **Balance** field. This field gives the area of each province in square kilometers. Change the number of **Districts** to **4** and press **OK**.

The Districts command will create a new column called **Districts** and will assign each province in *mexico* to one of four different districts by placing the value 1, 2, 3 or 4 into that column for each province. Thematic formatting will then be applied using the **Districts** column as a guide for formatting. The objective of the Districts command is to assign each province into districts so that the sum of the total **SQKM** for each district is the same and so that all areas in each district are contiguous (within the threshold parameter distance). To achieve districts where each district has the same overall **SQKM** total can require some oddly shaped districts although the Districts command will try to keep districts as compact as possible.
We can see that in the case of an irregularly shaped country such as Mexico it is a challenge for the Districts solver to create compact districts that also balance the desired field. In particular, we can select one province that seems it could be swapped for a different province.

Opening the drawing's table we can see that the province of Zacatecas has an area of 73252 square kilometers. It has been assigned to district 2, the brown district. [We used Table - Columns to show only three columns in the table.]

We can click on another province in the green district that seems it could be swapped for Zacatecas.
We see in the table that the province of **Guerrero** has an area of **64281** square kilometers. It has been assigned to district **1**, the green district. Swapping this province for **Zacatecas** would reduce the total square kilometers in the brown district and increase the total square kilometers of the green district, but not by much. We might decide that it is a reasonable trade to keep the provinces more compact.

We can reassign the **Guerrero** province to district **2** by double clicking into the **Districts** cell for the **Guerrero** record and entering the value **2** as seen above. We can then change the **Districts** cell for the **Zacatecas** record to **1** (not illustrated).

We can then click on the drawing, click on the background color well in the formatting toolbar, click on **Thematic** and reapply the formatting to update the formatting of the provinces to using the new values just entered for **District** in the toolbar. The result as seen above is a more compact allocation of provinces into districts.

### Performance

Districting is easy to do if a drawing fulfills certain limiting conditions, such as a guarantee that areas never overlap and are always perfectly adjacent. Since "real world" drawings often have imperfections, the Manifold districting algorithm uses a **Threshold** factor for determining distance between areas. The algorithm also uses a variety of computational geometry and network algorithms to deal with common imperfections in data sets. As a result, computation with a large number of areas could be slow with a large number of areas and a slow machine. To assess the performance of a particular machine with a given data set, begin by applying Districts to small data sets such as the sample **mexico_eg** drawing and scale up gradually.
The Districts command can create up to one million districts although, of course, it is normally used with a far smaller number of districts, since many districts are not easily comprehended.

See Also

Districts (Advanced)
Districts (Visual)
Orthogonalize

The Orthogonalize command moves coordinates to orthogonal grid positions at the specified X and Y grid steps. The command is used mainly to trim insignificant digits from coordinates. For example, suppose we are working with a projected data set expressed in meter-based coordinates and we know that our data set is only accurate within 10 meters. In such cases, there is only an illusion of accuracy when using coordinates such as 34592.490593845 meters. It may as well be 34590 meters. If every coordinate were moved to the nearest ten-meter coordinate position, we would achieve the effect of rounding all coordinates. Orthogonalize command performs exactly this function.

Scope  The set of objects to alter.
Steps  The X and Y dimensions of the grid cells.
Offsets  The X and Y offsets of a grid cell that may be used.

Offsets are combined with Steps to nudge the orthogonalization in desired X and/or Y directions. With X and Y Steps equal to 100 using zero Offsets will force the coordinates of objects to the nearest locations on a 100 x 100 grid of cells (for example, 0:0, 100:0, 0:100, ...).

With X and Y Steps equal to 100 using Offsets of 10 and 20 will force the coordinates of objects to the nearest locations on a 100 x 100 grid that is shifted in both horizontal and vertical directions by 10 and 20 units (for example, 10:20, 110:20, 10:120, ...).

Example

Suppose we start with a drawing of Durango province in Mexico using Orthographic projection.

After applying the Orthogonalize command with X and Y steps of 25000 meters the province takes on a distinctly stair-step appearance as coordinates are forced to the nearest 25000-meter grid position. The Orthogonalize command is not normally used in cases where the original coordinates appear in much finer resolution than the steps specified, so some regions have been collapsed into topologically redundant appendages. These can be fixed by running the Normalize Topology transform toolbar operator.
The Normalize topology operator cleans up the topology.
Segmentize
The Segmentize command adds redundant coordinates to objects. It is normally used to make ready for projection very large, simple objects that would otherwise suffer distortion. See the Segmentization topic for discussion of segmentization and projections. The Segmentize menu command applies additional coordinates so there is no straight segment of an object that has greater than the given Distance between coordinates.

**Scope**  The set of objects to alter.

**Distance**  Maximum distance between coordinates.

**Remove duplicate coordinates**  If checked (the default), removes duplicate coordinates. If unchecked, leaves duplicate coordinates intact.

**Example**

Suppose we have an area that is a simplified rendering of the province of Durango in Mexico, seen in Orthographic projection in a drawing. The province is approximately 440 kilometers wide. The simplified rendition was created by applying Simplify with a Distance setting of 25000 meters.

If we click on the area to select it as the primary selected object for editing, edit handles appear at each coordinate that define the area.
Zooming in to the Eastern side of the province we can see that the area consists of straight line segments between the coordinates that define the area.

If we apply Segmentize with a Distance setting of 10000 meters, Manifold will place redundant coordinates along the straight segments at a distance not exceeding 10,000 meters. Additional edit handles will appear for each coordinate.

**Transform Toolbar Segments Command**

The Segments command in the Transform Toolbar adds the specified number of segments, whereas the Segmentize command in the Drawing menu adds segments so that each segment is no greater than a given Distance parameter. For example, segmentizing a 100 meter line into 40 meter segments will create segments of 40, 40 and 20 meters.

**Tech Tips**

In some circumstances, Segmentize and Simplify are inverse functions. For example, if we applied Simplify again with a setting of 25000 after the Segmentize command shown above the redundant coordinates would be removed.

Many transforms automatically run a Normalize Topology transform before running to eliminate common errors that may affect correct operation. The Boundaries, Points and Node Points transform operators do not normalize target objects before running. This makes it possible, for example, to segmentize lines with a length threshold of 10 meters, and then use the Points transform to place point objects at the locations of resulting inflection points.

**See Also**

Segmentization
Segments
Simplify
**Simplify**

The **Simplify** command reduces the number of coordinates that define an object. Given a scope of action ([All Objects] or a selection) and a Distance, the **Simplify** command will reduce the number of coordinates such that the coordinates that define the object are approximately the given distance apart. The **Remove small branches** option will automatically remove branches in multi-branched objects (such as areas with islands or holes) that are smaller than the given distance.

In the example above, the province of Durango in Mexico has been simplified using **Simplify**. The drawing was first projected into Orthographic projection so that meter-based distances could be used.

The **Simplify** dialog was run with a Distance of 25000 meters.

**Comparison to Normalize Topology**

The **Simplify** command is similar to the Normalize Topology transform toolbar operator. However, the **Simplify** command is aimed at simplification on a per-object basis while the Normalize Topology operator considers relationships between objects when simplifying them. We can see the difference by running both commands on a drawing of Mexico.

We begin with a drawing of Mexico that's been projected into Orthographic projection.
Running **Simplify** with a distance setting of **25000** meters results in the simplification of areas. However, each area is simplified without considering any relationships with adjacent areas. This provides an optimal simplification when each area is considered by itself but also results in overlaps and gaps between areas.

If instead of using **Simplify** we used **Normalize Topology**, we would first use **View - Properties - Precision** to set the **Location Precision** to **25000** meters and then we would run **Normalize Topology**.

**Normalize Topology** takes a lot longer to run than **Simplify**, but the result in the end is free of overlaps and gaps. Each area has been adjusted both on the basis of capturing its own shape as well as to match its neighbors. However, it could be said that the resulting shape for each individual object (without regard to neighboring areas) is not as "optimal" a simplification as is done by **Simplify**.

When simplifying a single object it is probably best to use **Simplify** since the result is obtained much faster. When simplifying many objects for which adjacency must be maintained it is best to use **Normalize Topology**.

**See Also**

- Normalize Topology
- Segmentize
Spatial Overlay

Spatial overlays are a set of methods for transferring data between objects in drawings based on their spatial relationships to each other.

For example, suppose we have a drawing that shows states as areas and we also have a set of points showing the locations of cities. Suppose that each city record has a field called Population that gives the population in that city, but that we have no values for the population of each state. We could use spatial overlays to automatically add up the values of the populations for all cities in each state and place that combined value into the Population field for each state.

Spatial overlays transfer data from fields in a Source object set to a Target object set using some method. The specific rules used to transfer data are taken from the Transfer Rules specified for each column.

Spatial Overlay Dialog Controls

Source
Choose an object set to be the source of data. Choices will include the selection, all saved selections, and objects in drawing layers when launched from a map. The source set may contain only objects of one type (that is, all areas, all lines or all points).

Target
Choose an object set to receive the transferred data. Choices will include the selection, all saved selections, and objects in drawing layers when launched from a map. The target set may contain only objects of one type (that is, all areas, all lines or all points).

Method
Choose a transfer method. Only those operations that make sense for the objects in the source and target sets will be presented. If the source or target set do not contain all objects of one type the method box will not be enabled.

Spatial Overlay Methods

The following methods will appear in the Methods box as appropriate to the types of objects that have been selected in the Source and Target sets.

Areas to contained areas
Transfer fields from area A to areas that are completely inside A.

Areas to contained lines
Transfer fields from area A to lines that are completely inside A.

Areas to contained points
Transfer fields from area A to points inside A.

Areas to containing areas
Transfer fields from area A to areas that completely contain A.

Areas to intersecting areas
Transfer fields from area A to areas that have at least some location within A’s interior (that is not on a boundary).

Areas to intersecting lines
Transfer fields from area A to lines that have at least some location within A’s interior (that is not on a boundary). Excludes fully contained lines.

Areas to neighbor areas
Transfer fields from area A to areas that touch A only at A’s boundary (that is, have no common locations that are interior for A).

Areas to neighbor lines
Transfer fields from area A to lines that touch A only at A’s boundary.

Areas to boundary
Transfer fields from area A to points that lie on its
<table>
<thead>
<tr>
<th>Transfer Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Points to containing areas</td>
<td>Transfer fields from point P to areas that contain P.</td>
</tr>
<tr>
<td>Points to containing lines</td>
<td>Transfer fields from point P to lines that contain P.</td>
</tr>
<tr>
<td>Points to neighbor areas</td>
<td>Transfer fields from point P to areas that contain P on their boundary.</td>
</tr>
<tr>
<td>Points to neighbor lines</td>
<td>Transfer fields from point P to lines that contain P on their boundary.</td>
</tr>
<tr>
<td>Points to coinciding points</td>
<td>Transfer fields from point P to points that coincide with P.</td>
</tr>
<tr>
<td>Lines to contained areas</td>
<td>Transfer fields from line L to areas that are completely contained by L.</td>
</tr>
<tr>
<td>Lines to contained lines</td>
<td>Transfer fields from line L to lines that are completely contained by L.</td>
</tr>
<tr>
<td>Lines to intersecting areas</td>
<td>Transfer fields from line L to areas that share at least one interior location with L.</td>
</tr>
<tr>
<td>Lines to intersecting lines</td>
<td>Transfer fields from line L to lines that share at least one interior location (that is, a location that is not a terminal coordinate of any branch) with L.</td>
</tr>
<tr>
<td>Lines to neighbor areas</td>
<td>Transfer fields from line L to areas that are touched by L at their boundary.</td>
</tr>
<tr>
<td>Lines to neighbor lines</td>
<td>Transfer fields from line L to lines that touch (and are touched by) L at the terminal coordinates of any L branch.</td>
</tr>
<tr>
<td>Lines to terminal points</td>
<td>Transfer fields from line L to points that lie on its boundary (that is, terminal points of branches).</td>
</tr>
<tr>
<td>Lines to touching areas</td>
<td>Transfer fields from line L to areas that share at least one location with the line.</td>
</tr>
<tr>
<td>Lines to touching lines</td>
<td>Transfer fields from line L to lines that share at least one location with L.</td>
</tr>
</tbody>
</table>

**Example**

Let's consider a simple example to see how spatial overlays work.
We've created a drawing with two areas and nine points. Select the points and save them as a saved selection called **Points** in the Selections pane. Select the two areas and save them as a selection called **Areas** in the Selections pane.

The table for this drawing shows that there is one integer field, **Population**. Each of the points has a value of **1** for the population field and is shown with red selection background color in the table. The two areas have values of **0** for the population.

Open the drawing’s table, right click onto the **Population** column and choose **Transfer Rules**.

In the Transfer Rules dialog choose **Copy** as the transfer rule for **1 to N** transfers and choose **Sum** as the transfer rule for **N to 1** transfers.

Click onto the drawing window and choose **Drawing - Spatial Overlay**.

In the Spatial Overlay dialog choose **Points** as the Source and **Areas** as the Target. The Method will be **Points to containing areas**. Press **OK**.
The result of the spatial overlay operation is that one area (the triangle) acquires a value of 2 and the other area (the circle) acquires a value of 4. These new values are the sums of the point values within the areas.

**Geographic Example**

Let's apply the above procedure in a geographic setting. We will sum up the populations of towns to get a total population for each county containing the towns.

We have a drawing called *Example* that shows counties in the San Francisco Bay area as areas together with points taken from a Census Bureau file of named places with populations.

<table>
<thead>
<tr>
<th>Example Table</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type (t)</strong></td>
<td><strong>county</strong></td>
<td><strong>Name</strong></td>
</tr>
<tr>
<td>Area</td>
<td>Napa</td>
<td>0</td>
</tr>
<tr>
<td>Area</td>
<td>Sonoma</td>
<td>0</td>
</tr>
<tr>
<td>Area</td>
<td>Solano</td>
<td>0</td>
</tr>
<tr>
<td>Area</td>
<td>Marin</td>
<td>0</td>
</tr>
<tr>
<td>Area</td>
<td>Contra Costa</td>
<td>0</td>
</tr>
<tr>
<td>Area</td>
<td>Alameda</td>
<td>0</td>
</tr>
<tr>
<td>Area</td>
<td>San Francisco</td>
<td>0</td>
</tr>
<tr>
<td>Area</td>
<td>San Mateo</td>
<td>0</td>
</tr>
<tr>
<td>Area</td>
<td>Santa Clara</td>
<td>0</td>
</tr>
<tr>
<td>Area</td>
<td>Santa Cruz</td>
<td>0</td>
</tr>
<tr>
<td>Point</td>
<td>Alamo</td>
<td>... 12277</td>
</tr>
<tr>
<td>Point</td>
<td>Angelina</td>
<td>... 3503</td>
</tr>
<tr>
<td>Point</td>
<td>Antioch</td>
<td>... 62195</td>
</tr>
<tr>
<td>Point</td>
<td>Antioch</td>
<td>... 9654</td>
</tr>
</tbody>
</table>

Opening the *Example Table* we see that there are ten areas each with a 0 value in the *pop1990* field. Numerous towns have populations for each town.

We begin by right clicking on the *pop1990* column header in the table and choosing *Transfer Rules.*
We set the transfer rules for this field to be \textbf{Sum} for \textbf{N to 1}. Since many points will be combined into each county this is a "many to 1" or \textbf{N to 1} transaction. We would like the population values to be summed to get the value for each county.

We don't want any values transferred for the \textbf{county} or \textbf{Name} fields. We right click onto the \textbf{county} column header, choose \textbf{Transfer Rules} and choose \textbf{None} for the \textbf{N to 1} transfer rule. We then right click onto the \textbf{Name} column header, choose \textbf{Transfer Rules} and once more choose \textbf{None} for the \textbf{N to 1} transfer rule.

If we were using spatial overlays between different drawing layers in a map we could skip this step; however, since we are transferring values between fields in the same drawing we will use saved selections. Select all points in the drawing and save this selection as a saved selection in the Selections pane called \textbf{Places}. Select all points in the drawing and save this selection as a saved selection called \textbf{Counties}.

With the focus on the \textbf{Examples} drawing window, choose \textbf{Drawing - Spatial Overlay}.

In the \textbf{Spatial Overlay} dialog choose \textbf{Places} for \textbf{Source} and \textbf{Counties} for \textbf{Target}. For the \textbf{Method} choose \textbf{Points to containing areas} and press \textbf{OK}. 
In the table we can see that the `pop1990` values are filled with the sum of the point values for each county. This is in accordance with the `N to 1` transfer rule we specified for this field. Note that there is no change in the `county` or `Name` fields for the county areas since we specified `None` as the transfer rules for these fields.

### Spatial Overlays in Maps

Spatial overlays may be used in maps when a map contains at least one non-empty drawing. To use spatial overlays between different drawings in a map, the following must be true:

- The same field name and type must be present in both drawings involved.
- Exactly the same set of transfer rules must be used in both drawings for each field involved.

If the above two conditions are not met for a particular field it will not be transferred at all.

Note that the "same type" requirement is meant literally: for example, fixed length ASCII text columns of different length are different types. Therefore, attempting to transfer data between a 20-character fixed length ANSI text column and a 40-character fixed length ANSI text column will not work due to the different lengths of the data allowed.

Note: A linked drawing or a shared drawing in Enterprise Edition that is not checked out itself but has a checked out table can nonetheless be the target of a spatial overlay.

### Troubleshooting

Difficulties with spatial overlays arise almost exclusively from failing to set Transfer Rules correctly for the desired action. For example, if the `Transfer column` checkbox in the `Transfer Rules` dialog has been unchecked a column will not transfer. If no columns are available for transfer the spatial overlay operation will report "No columns to transfer."

### See Also

Transfer Rules - Transfer rules specify how fields are aggregated or allotted when new objects are created using transform toolbar operators. Spatial overlays are a different concept than the idea of using transfer rules to aggregate or to allot values when creating objects with transforms. However, it is a related concept in that aggregations or allotments are used to transfer field values between objects.

Transfer Contour Line Height to Points - A simple example using Spatial Overlay.

Color Areas by Counts - A simple example using Spatial Overlay to color areas by the number of points they contain.

Shortest Path over Land - A complex example that uses Spatial Overlay.
Intersection Overlays - A small, but sophisticated, example that uses Transfer Rules, Split with, Clip with (Intersect) and Spatial Overlay commands with areas.
Topology Overlay

The Topology Overlay dialog creates a results drawing from two other drawings using Identity, Intersect, Union and Update overlay functions. These functions may also be performed by a sequence of transform toolbar operators. The Topology Overlay tool provides one-step functionality to make it easy to perform these common functions. Overlays simultaneously create new objects in the results drawing and also populate data columns as necessary. Overlay functions are also referred to simply as overlays.

All overlays require two drawings: a data drawing that may contain areas, lines and points and an overlay drawing that must contain areas. The areas in the overlay drawing guide the operation of the chosen overlay function to create a new, result drawing.

The result drawing will inherit all columns from both the overlay drawing and the data drawing. There is no mapping between the columns in the overlay drawing and the data drawing. Each resulting object inherits all attribute values from the overlay drawing and from objects in the data drawing from which it has been produced. A topology overlay will automatically copy over values from active columns, rank columns, and columns brought into a drawing's table via relations.

- **Identity** splits all areas, lines and points in the data drawing with areas in the overlay drawing and places each resulting part of the original object from the data drawing into the result drawing.
- **Intersect** intersects all areas in the data drawing with areas in the overlay drawing and places each resulting part of the original area from the data drawing that lies in at least one area in the overlay drawing into the result drawing.
- **Union** intersects all areas in the data drawing with areas in the overlay drawing and places each resulting part of the original area from the data drawing into the result drawing, then it intersects all areas in the overlay drawing with areas in the data drawing and places each part of the original overlay area that does not lie in any area in the data drawing into the result drawing.
- **Update** intersects all areas, lines and points in the data drawing with areas in the overlay drawing and first places each resulting part of the original object from the data drawing that does not lie in any overlay area into the result drawing, then second placing each overlay area into the result drawing.

The above overlays are similar to those in various legacy GIS systems, except that they are not subject to some of the limitations that these GIS systems have. In particular, Manifold allows the data drawing to contain areas, lines and points (and not just areas as is the case with some systems) and overlays these objects simultaneously without forcing the user to perform the overlay operation once for each object type as would be required by some other systems. Manifold also automatically resolves overlaps between areas and other topological anomalies both in the data drawing and the overlay drawing, without forcing the user to go through a separate topology cleaning step.

Performing a Topology Overlay

1. Open the drawing to be used as the overlay drawing or data drawing, or open a map containing the drawing and set the drawing's layer active.
3. In the Topology Overlay dialog, choose the Data drawing, the Overlay drawing and the Method. Press OK.

Topology Overlay Dialog Controls

When choosing object sets to use in the Overlay or the Data drawing, choices available will be all objects or any selection or saved selection in that drawing.

- **Data** Choose an object set in the data drawing upon which the overlay will operate.
- **Overlay** Choose an object set in the overlay drawing that will control the overlay. Only areas can be used to control the overlay.
- **Method** Choose an overlay method.
- **Force copy / copy transfer rules** When checked, ignores the transfer rules defined for each column and instead always applies a copy / copy transfer rule. Similar to how legacy GIS applications...
operate. Not checked by default.

**Topology Overlay Methods**

The following methods will appear in the **Method** box for topology overlays:

- **Identity** Split all areas, lines and points in the data drawing with areas in the overlay drawing and places each resulting part of the original object from the data drawing into the result drawing.

- **Intersect** Intersect all areas in the data drawing with areas in the overlay drawing and place each resulting part of the original area from the data drawing that lies in at least one area in the overlay drawing into the result drawing.

- **Union** Intersect all areas in the data drawing with areas in the overlay drawing and place each resulting part of the original area from the data drawing into the result drawing. Next, intersect all areas in the overlay drawing with areas in the data drawing and place each part of the original overlay area that does not lie in any data area into the result drawing.

- **Update** Intersect all areas, lines and points in the data drawing with areas in the overlay drawing and first place each resulting part of the original object from the data drawing that does not lie in any overlay area into the result drawing, then second place each overlay area into the result drawing.

**Examples**

Let's consider two drawings, **A** and **B** and the result of overlays between them.

![Image of a drawing with areas](image)

**Drawing A** consists of two areas.
Drawing B contains three areas.

If we show A and B together in a map we can see that the areas in A intersect the areas in B.

The following illustrations show the results of an "A operation B" where B is the Overlay drawing and A is the Data drawing. The results of each operation are shown as blue areas in a third results drawing shown in a layer above both the A and B drawings.

For example, we can use the Identity method with all objects in A as the Data drawing and all objects in B as the Overlay drawing.

Identity creates a result set of all the areas in A, but split up by their intersections with the areas in B. Instead of two areas in A, the result is eight areas in A. Each original A area has been split by its intersection with the three B areas to create three smaller areas plus a fourth area representing that part of the original A area not intersecting any B areas.
**Intersect** creates six areas in the result set, three areas from both of the original A areas where they intersect the B areas.

**Union** chops up all areas by their intersections between areas in A and B. It is, in effect, a dual “cookie cutter” operation performed using B areas to cut A areas and then using the results to cut B areas as well.

**Update** preserves all B areas and moves them into the result. It takes that part of any A area not lying within an intersection with a B area and moves it into the result as well. We can think of **Update** as being an “updating” of any changes in A areas using B areas (the **Overlay**) as being an overriding control. For example, suppose we have some town boundaries that have been changed but we want areas that represent unchanged water bodies (such as lakes or ponds) to be preserved in any intersection of water bodies and new town boundaries. In that case we could make our changes in town boundaries and then do an **Update** using the water areas as the **Overlay** and know that whatever changes were made in town boundaries the water areas would “cookie cut” through any boundary areas as necessary so that water areas would remain invariant.

**See Also**

Traditional Topology Tools - An optional add-in that may be downloaded from the manifold.net web site. This add-in provides operations using stored topology.
Traditional Topology Tools

The **Traditional Topology Tools** add-in may be downloaded from the [manifold.net](http://manifold.net) web site. If the add-in is not installed the capabilities in this topic will not be available. This add-in is often called **Topology Tools** for short. The add-in provides topology tools that employ stored topology using the methods of, and requiring constraints similar to, those of legacy GIS packages. In addition to being useful in some cases in expert hands, the tools also provide an interesting glimpse into the relative performance attainable in different cases using an "apples to apples" comparison between different topology computation approaches.

The add-in is published in 32-bit or 64-bit versions. Download and install the 32-bit version if you are running 32-bit Manifold System. Download and install the 64-bit version if you are running 64-bit Manifold System. Install the add-in by double-clicking on the .msi installation package.

**Using the add-in:**

1. Open a .map file that contains drawings. If any editing or changes have been made, save the project.
2. Choose **Tools - Add-Ins - Traditional Topology Tools** to launch the add-in.
3. Choose the operation desired and the drawings to be used as the initial drawing and overlay.
4. Press the **Perform Operation** button. The bottom pane in the dialog will record a running history of actions taken. Result drawings will be created in the project pane listing.

**Overview**

Unlike older GIS packages, Manifold does not build topological relationships into the data set: rather, topological relationships are computed on the fly as needed.

Years ago when computers were weak it made good sense to embed topological relationships into the data sets themselves. In modern times it is often better to compute topology on the fly. Doing so uses the intelligence of the machine to free the user from having to worry about low-level structural details. It allows us to mix points, lines and areas in the same drawing and to move objects about interactively with great freedom.

There are cases, however, when the use of stored topology will provide faster performance if we are willing to live within the constraints of this traditional method. For those users who have the expertise and the interest in using traditional methods, the Traditional Topology Tools add-in provides an optional toolset implementing those traditional methods. This add-in has been released on an informal basis. No technical support is available for this add-in.

The toolset is implemented as an add-in and not as a built-in part of Manifold because the usage of stored topology is very different from the topology on the fly approach used within Manifold. Using this add-in requires expert level understanding of the stored topology approach, which will probably have been acquired during work with a legacy system.

The add-in is not recommended for use by beginners. In particular, beginners in search of higher performance should not reach for this add-in: instead, they should study carefully the Performance Tips topic to learn how to assure maximum performance in Manifold operations. Although this add-in will increase performance in some cases, it will only do so when performing one of the four topology overlay operations available in the Topology Overlay dialog or in this add in. Using this add-in will not change the performance of any other Manifold operation. Even in those cases where this add-in may perform faster than the standard Topology Overlay dialog, it is unlikely any performance gain will be significant enough to be noticed unless the drawings involved are very large.

The add-in includes two basic types of functions:

- **Clean Topology** - Normalizes the topology of a given drawing and creates a "clean" version of the drawing as is often done within legacy GIS packages. The closest equivalent in regular Manifold usage is the Normalize Topology transform operator.

- **Overlays** - Four functions to perform **Identity, Intersect, Union** and **Update** overlays. The overlay functions will overlay a pair of drawings and create a result drawing. The closest equivalent in regular Manifold usage is the Topology Overlay dialog, which provides equivalent operations implemented using standard Manifold topology on the fly. See the Topology Overlay dialog for illustrations of these four operations.
Constraints

The functions operate like their counterparts in legacy GIS products. As a result, they have a number of limitations just like those in legacy products:

- Functions can only operate on entire drawings. They cannot operate using only a selection.
- Functions cannot alter existing drawings. The results must be stored in a new result drawing.
- Functions do not use transfer rules - they always copy data.
- Functions can only operate on objects of the same type. For example, they will ignore lines and points in a drawing that contains areas.
- Functions do not allow areas to overlap. If a drawing contains areas that overlap, every region that is covered by more than one area will be assigned to one of those overlapping areas for the purposes of the overlay function.

Controls

<table>
<thead>
<tr>
<th>Operation</th>
<th>Choose from Clean Topology, Identity Overlay, Intersect Overlay, Union Overlay or Update Overlay.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drawing</td>
<td>Target drawing. Only this box is enabled for Clean Topology.</td>
</tr>
<tr>
<td>Overlay</td>
<td>Drawing that controls the overlay.</td>
</tr>
<tr>
<td>Perform Operation</td>
<td>Press to apply the operation.</td>
</tr>
<tr>
<td>Save topology data on disk</td>
<td>Cache stored topology on disk for maximum performance.</td>
</tr>
</tbody>
</table>

"Save topology data on disk" Option

For maximum performance, turn on the Save topology data on disk option, which tells the tool to cache topology structures on disk. With this option turned on, topology structures will be computed and saved in a separate file on disk for each drawing. Each drawing will have topology stored in a separate file named map.drawing-id.topology where map is the name of the .map file and id is the ID of the drawing within that .map file.

The size of topology data in the cache file will be approximately the same as the size of the producing drawing, excluding the size of the drawing's table. The system keeps track of changes made to the drawing and will not use outdated topology structures. If the .map file has not yet been saved, the option is disabled and the system can neither save topology data nor use already saved data.

One other limitation when caching topology data is that the coordinate systems and location precision parameters of overlaid drawings must be exactly the same. If either of these two characteristics are different, the tool will not be able to use cached topology data and will have to generate the topology of an overlaid drawings on the fly.

As a practical matter, therefore, operating the add-in without checking this option will result in performance not much different from standard topology on the fly. To take advantage of stored topology, check the Save topology data on disk option to turn on caching, do a Clean Topology operation on each drawing to be used to created stored topology, and then run the overlay operation desired. When drawings are selected in the Drawing and Overlay boxes, Manifold will report if the drawing has stored topology available in cache.

Example

Consider a project that contains two drawings, called A and B.
The illustration above shows both drawings together in a map. Drawing A contains rectangular areas formatted in blue. Drawing B contains circles formatted in yellow. The B drawing layer in the map uses partial opacity so that overlapping regions are clearly visible.

We launch the add-in and choose the **Intersect Overlay** operation, using A as the drawing and B as the overlay. The result is placed in a drawing called **A - Intersect - B**.

We can drag and drop the resulting drawing into the map to see how the objects it contains arise from the intersection of areas in the A and B drawings. Areas in the **A - Intersect - B** drawing have been formatted in magenta color.
If we click off the \textit{A} and \textit{B} layers we can see clearly the area objects in the new \textit{A - Intersect - B} drawing.

**Notes**

The Traditional Topology Tools add-in arises from experimental work done at \texttt{manifold.net} to quantify the performance differences between topology on the fly and stored topology. They have been released on an informal basis to allow users to experiment with and apply stored topology computation in cases where the user judges such computation to be advantageous.

Stored topology offers performance advantages when computing topology overlays for large data sets on weaker computer systems, but at the cost of significant limitations compared to topology on the fly. As computer systems get faster and Manifold topology on the fly computational algorithms get faster, it becomes possible to do topology on the fly even with larger data sets at a speed that equals or at times exceeds the use of stored topology. Already in most, but not all, cases topology on the fly is as fast or faster than using stored topology.

Stored topology is an effective approach for single-threaded operation but it is very poorly suited to multi-threaded, parallel or distributed operations using multiple processors. In contrast, topology on the fly is a very good approach for parallel or distributed code. Processors already feature dual cores or quad cores. As parallelization on the desktop increases, very rapidly the time will come when suitably parallelized topology on the fly algorithms will always exceed the speed of stored topology even when drawings containing very many objects are being overlaid.

This add-in has been released on an informal basis for users who wish to apply traditional methods for topology overlays. However, as computer systems continue to improve and desktop architectures become multi-core by default, it is expected that these tools will no longer be provided because they will be obsolete even as an informal, interim experiment. Therefore, any use of these tools should be conditioned on the understanding that they represent an experimental, interim toolset that may change without notice or at some time no longer be available.

**See Also**

Normalize Topology

Topology Overlay
The Transform toolbar makes changes throughout the entire drawing using the specified operators. Transform operators can create new objects, delete objects, change objects (for example, splitting them) and select objects using various algorithms such as the location of a shortest path. The transform toolbar is also used to make spatial selections using commands such as Select Touching or Select Contained within that work between sets of objects.

The Transform toolbar consists of three boxes, from left to right: a **target box**, the **operation box**, and a **source / argument box**.

**Target box**  Also known as the **scope box**. The objects that will be affected, altered or which will control the operation. Choices in the target box will be [All Objects], [Selection] or the names of any saved selections that have been saved in the Selections pane. The illustration above shows that the operator will be applied to all objects in the drawing.

**Operation box**  The function to be applied. The operation box is context sensitive and will show only those operations that make sense for drawings. The example shows we will add border buffer zones.

**Source / argument box**  The value to be used. Depending on the operator, this may be another object set or a value entered by the user. Many operators, such as Boundaries do not require a source or argument. The source / argument box will not be enabled for such operators. In the example above we use 10000 for the size of the border buffer zones, the size being specified in the native units of the drawing.

We use dark blue, black, and violet bold face fonts in this topic to distinguish the three boxes to make examples and explanation more clear. In real life Manifold uses the same black font color in all three Transform toolbar boxes.

**Using the Transform toolbar**

1. Click on the map layer or drawing that contains the target objects.
2. Make a selection if the operation is to be applied just to the selection and choose [Selection] in the target box. Alternately, choose the name of a previously saved selection.
3. Choose the desired operator in the operation box.
4. Choose or specify a value in the source / argument box, if this operation requires it.
5. Press **Apply**.
6. Some operators create new objects. Any new objects created will be created in the active drawing/layer and will now be the Selection. Move the Selection to a new layer/drawing if you wish to keep these objects organized separately from the original objects in the drawing (almost always a good idea).

**Target box**

The left-most box specifies the **target objects**. These are the objects that will be affected, altered or which will control the operation. The example above shows that there is no selection present, so the Transform toolbar will apply the Transform function to all of the objects in the drawing.
If we make a selection, we can switch the target box to [Selection] to apply the operation to only those objects that are part of the current selection. We could also choose the name of a saved selection.

**Operation box**

Choose a function from the long list of operators available in the operation box. The example shows **Boundaries**, which will create lines along the periphery of each area. The Transform Operators - Drawings topic lists operators available for use with drawings.

**Source / argument box**

Many operators do not require any sources or arguments. For these functions the source / argument box will be disabled.

Other functions will require specification of a source. This may an object set such as [All Objects], [Selection] or the names of saved selections, all of which can be chosen from the combo box.

Some functions require an argument value that is entered by the user. To enter a value, click into the source / argument box, enter the value using the keyboard and then press **Enter**.

**Examples**

[All Objects] **Boundaries** - Creates boundary lines on the periphery of all areas (if any) in the drawing.

[Selection] **Boundaries** - Creates boundary lines on the periphery of all areas (if any) that are in the current selection in the drawing.

[Selection] **Split with Roads** - Take all objects in the saved selection named Roads and split each selected object using these objects.

The selection pane is often very important when working with the transform toolbar. Saved selections can appear in either the target box or the source / argument box. See the Selection topic details on using the selection pane.

**Saved Selections in Maps**

When drawings appear together in a map, if any drawing layer has the focus the transform toolbar for the map will list all saved selections in all drawings. This is a great convenience, but it also leads to a slight complication in that all saved selections using the same name are treated alike. Saved selections in the same drawing are required to have different names, but saved selections in different drawings might use the same name.
The transform toolbar for the map will treat all drawing saved selections using the same name as one, combined saved selection. For example, if we have two drawing layers in a map, one called East and the other called West and they both have a saved selection called Cities, then using the transform toolbar to create a convex hull using the saved selection choice called Cities will use the objects from both the East and West drawings that appear in their respective Cities saved selection.

Therefore, it is a good idea to use unique names for saved selections in drawings that might appear together in the same map. For example, we could name the saved selection in one drawing East - Cities and that in the other drawing West - Cities.

**See Also**

Transform Operators - Drawings for a detailed list of operators available.
Transfer Rules for information on controlling how database table information is transferred to new objects that are created by transform operators. For example, when creating centroids the transfer rules within the drawing's table specify which fields should be transferred to the centroids.
Transform Operators - Drawings

The Transform toolbar for drawings includes a wide variety of operators. Some are specialized operators used to repair topology or to prepare drawings for use as networks in network or transportation analysis while other operators are more commonly used in GIS. The hyperlinks below will jump to the Help topic for the corresponding command.

**Attach to / Attach to Self**
These operators move the coordinates of one object to align to the coordinates of another object or objects. They are mainly used to adjust areas and lines to the shapes of adjacent areas or lines within the same object set or across different object sets.

**Border Buffers**
Create a buffer zone for each area object extending both outward and inward from the area's boundary the given distance.

**Boundaries**
Create lines in the shape of the periphery for each area in the target set. Will create duplicate boundary lines in places where the edges of areas coincide.

**Bounded Areas**
Create areas within regions enclosed by lines in the target set.

**Bounding Boxes**
Create an area in the form of a rectangular bounding box that encloses each object and is aligned vertically and horizontally with the map window.

**Buffers**
Create a buffer zone for each object extending outward by the given distance. For areas, this is an outer buffer zone.

**Centroids**
Create a point at the center of the minimum circle that encloses each object. (areas or lines)

**Centroids (Box)**
Create a point at the center of the minimum enclosing rectangle for each object. (areas or lines)

**Centroids (Inner)**
Create a point at the "center" of an area and adjust the position of the point so that it always falls within the area. (areas only)

**Centroids (Weight)**
Create a point at the approximate center of balance of each object. (areas only)

**Clip with (Intersect)**
Use the areas in the source / argument box set to "cookie cutter" objects in the target set and leave only those items inside the cookie cutter.

**Clip with (Subtract)**
Use the areas in the source / argument box set to "cookie cutter" objects in the target set and remove those items inside the cookie cutter.

**Clusters**
Given a Parameter value from 0 to 100, detects clusters and draws lines between all points in the cluster. Exists in standard and Zahn algorithm versions.

**Common Bounding Box**
Like Bounding Boxes, but creates one bounding box about all objects in the scope.

**Common Buffer**
Like Buffers, but creates one buffer zone about all objects in the scope.

**Common Centroid**
Like Centroids, but creates one centroid using the center of the minimum enclosing circle for all objects in the scope.

**Common Enclosing Circle**
Like Enclosing Circles, but creates one minimum enclosing circle for all objects in the scope.

**Common Enclosing Rectangle**
Like Enclosing Rectangles, but creates one minimum enclosing rectangle for all objects in the scope.
Constrained Triangulation  A constrained Delaunay triangulation consisting of both lines and areas that does not cross line segments within the source object set.

Constrained Triangulation Areas  A constrained Delaunay triangulation consisting of areas that does not cross line segments within the source object set.

Constrained Triangulation Lines  A constrained Delaunay triangulation consisting of lines that does not cross line segments within the source object set.

Convex Hull  Draw an area that just exactly encloses all objects using the extremal coordinates as bounding coordinates for the area's boundary.

Decompose  Split branched objects into simple (non-branched) objects. For example, a single, branched area object composed of three "islands" will be split into three separate area objects.

Decompose to Convex Parts  Splits area objects into smaller areas automatically. Each smaller area created will be a convex hull for the locations it encloses. This operator is often used to split area objects into smaller areas to allow computations or to create centroids that fall within areas.

Decompose to Triangles  Automatically splits area objects into triangles.

Distance Network  Build a network that connects every pair of points that are within a given distance of each other.

Enclosing Circles  Create an area in the form of the minimum enclosing circle for each object.

Enclosing Rectangles  Create an area in the form of the minimum enclosing rectangle for each object, allowing the rectangle to be rotated as compared to the map window.

Explode  Explode lines into multiple line objects where each line segment from the original line becomes a separate line object. No effect on areas or points.

Farthest Neighbor  Assign each object in the Parameter set to the farthest object from it in the drawing and draw a line between them. Exists in standard and Symmetric versions.

Farthest Pair  Find the two objects that are farthest apart.

Flip Horizontally  Flip objects left / right to their mirror image.

Flip Vertically  Flip objects top / bottom to their upside down image.

Gabriel Network  Creates a Gabriel network on the target point set.

Inner Buffers  Create a buffer zone for each area that is within and smaller than the area by the given distance.

Intersect Lines  Split all lines in the target set into separate lines at intersections and place points at the intersections as well.

Intersection Points  Create points at the intersections of all lines in the target set, including self-intersections.

Join Lines  Join lines with coinciding ends into a new, single line object. Redistribute any data attribute fields according to transfer rules.

Move Horizontally  Move all objects or selected objects the specified distance horizontally. Negative values move objects to the left (West).
| **Move Vertically** | Move all objects or selected objects the specified distance vertically. Negative values move objects down (South). |
| **Nearest Neighbor** | Assign each object in the Parameter set to the nearest object to it in the drawing and draw a line between them. Exists in standard and Symmetric versions. |
| **Nearest Pair** | Find the two objects that are closest together. |
| **Node Points** | Create points at the terminal coordinates of lines, avoiding duplicates at locations where multiple lines terminate. |
| **Normalize Metric** | Fix common flaws in object metrics. |
| **Normalize Topology** | Fix common problems in topology, including snapping lines to nearest points, resolving intersections, resolving overshoots and undershoots and other miscellaneous topological anomalies based on settings in the View - Properties - Precision dialog. Also used to "generalize" drawings to lower precision. |
| **Points** | Create points at all coordinates defining the shape of lines and areas in the target set. Avoids making duplicate points in places where the edges of two areas coincide. |
| **Relative Neighborhood Network** | Relative neighborhood network. |
| **Remove Duplicates** | Remove duplicate objects. Deletes areas, lines or points that appear exactly at the same location. |
| **Reverse Lines** | Reverse the order of coordinates that define a line, thus reversing the "direction" of the line. Used with asymmetric formatting styles so that the style is oriented as desired. |
| **Rotate** | Rotate objects about the approximate center of their minimum enclosing circle by the number of degrees given in the source / argument box. Negative degree values cause counter-clockwise rotation while positive values cause clockwise rotation. Fractional values are allowed. |
| **Scale** | Resize objects by the scale given in the source / argument box. A parameter value of 0.5 resizes objects to half their size. A parameter value of 2 resizes objects to twice their size. |
| **Scale Horizontally** | Resize objects by the scale given in the source / argument box only in their horizontal (East / West) extent. A parameter value of 0.5 resizes objects to half their size. A parameter value of 2 resizes objects to twice their size. |
| **Scale Vertically** | Resize objects by the scale given in the source / argument box only in their vertical (North / South) extent. A parameter value of 0.5 resizes objects to half their size. A parameter value of 2 resizes objects to twice their size. |
| **Segments** | Add redundant coordinates to objects so they are defined using a larger number of coordinates (and thus, segments). This allows preservation of shape when projecting large, simple, vector shapes. |
| **Select Adjacent to** | Select all objects in the target set that are adjacent to any object in the source / argument box set. Objects are adjacent if they have one or more boundary coordinates in common with no other overlaps. |
| **Select Contained within** | Select all objects in the target set that are entirely contained within an object in the source / argument box set. |
| **Select Containing** | Select all objects in the target set that completely contain |
any object in the source / argument box set.

Select Critical Service Centers
Given a set of nodes, called clients, in a network finds the nodes that minimize the maximum path length to any of the clients. This is the classic problem for locating an emergency service center, where at any time one may be called upon to travel from the center to any client. The Select N Critical Service Centers transform uses the source / argument box to set the number of service centers allowed.

Select Euclidean Point Coverage
Select a minimal set of points so that each point from the source set is within the given distance (in the drawing's native measurement units) to at least one point of the result set.

Select Intersecting
Select all objects in the target set that intersect any object in the source / argument box set. Objects intersect if they have any parts in common excluding intersections consisting of common boundary coordinates only.

Select Line Coverage
Select a minimal set of lines so that each line from the source set touches at least one line from the result set.

Select Point Coverage
Given a network, selects a minimal set of points so that each point from the source set is adjacent (through the network) to at least one point of the result set. Normally used with [Selection] in the scope box.

Select Service Centers
Given a set of nodes, called clients, in a network finds the center nodes that minimize the path length to travel from the center to each of the clients and to then return to the center. This is the classic problem for locating a regular or daily service center, where every day one must make the rounds of each of the clients. The Select N Service Centers transform uses the source / argument box to set the number of service centers allowed.

Select Shortest Path
Finds the shortest path between two points in a network.

Select Spanning Tree
Given a network, select lines in the network that make up a minimum spanning tree.

Select Touching
Select all objects in the target set that touch any object in the source / argument box set. Objects touch if they either intersect or are adjacent to each other.

Shape Hull
Tile objects in the drawing with polygonal areas built up out of square tiles using the grid specified in the parameter.

Given a set of points, build a minimum spanning tree network on that point set by adding lines.

Spline
Smooths lines and areas by inserting additional midpoints and using a spline algorithm to replace sharp corners with curves.

Split With
Split target areas and lines using a given set of lines.

Triangulation
Create a Delaunay triangulation consisting of both lines and areas for the selected point set.

Triangulation Lines
Create a Delauney triangulation consisting of lines for the selected point set.

Triangulation Areas
Create a Delauney triangulation consisting of areas for the selected point set.

Union
Combine all areas in the target set into one area object.

Voronoi Diagram
Create area, line and point objects for each Voronoi cell.
Voronoi Lines  Create line objects for each Voronoi cell.

Voronoi Areas  Create area objects for the Voronoi diagram.

Voronoi Points  Create point objects for the Voronoi diagram.

**Automatic Normalization**

This is an advanced subject that assumes the reader is fluent with the internal topological structure of objects.

Manifold maintains automatic cleaning of object metrics. Automatic normalization occurs for those transform operators that require clean object metrics. Before any such Transform toolbar operator begins operation, Normalize Metric automatically will be run on all objects that need to be cleaned.

The automatic normalization is an internal Manifold function that occurs whenever it is required during transform toolbar operation. Due to the caching behavior of Normalize Metric the amount of time for cleaning (if required) will vary depending on whether or not objects in the drawing are marked "clean" or not. A particular transform operator might run very rapidly and then just after **Location Precision** has been changed suddenly run much more slowly during the next run. The slower run occurs because all of the objects have had their "clean" flag cleared as a result of the precision change, so Normalize Metric will have more work to do for that run.

When dealing with imported drawings that may contain specific features of the metric that are to be preserved, users will often choose to begin operations by running the Normalize Topology transform operator on all objects in the drawing. This will clean object metrics while preserving redundant coordinates that are co-located with other objects, and it will mark all objects as having clean metrics. This avoids a situation where Manifold is compelled to run Normalize Metric even though we do not want it run.

When in doubt run **Normalize Topology** once before running other transforms. This can be very time consuming for large drawings, but it is a prudent and necessary step experts will take if they wish to preserve specific metrics.

**See Also**

Transfer Rules for information on controlling how database table information is transferred to new objects that are created by transform operators. For example, when creating centroids the transfer rules within the drawing’s table specify which fields should be transferred to the centroids.
**Transform - Attach to / Attach to Self**

These operators move the coordinates of one object to align to the coordinates of another object or objects. They are mainly used to adjust areas and lines to the shapes of adjacent areas or lines within the same object set or across different object sets.

Both the Attach to and the Attach to Self operators adjust coordinates within the location precision distance parameter that is set for drawings in the View - Properties - Precision dialog. If the location precision parameter is larger than the distance between objects, the objects will be adjusted by moving the coordinate locations that define the objects to the nearest coordinate location of the reference object.

The Attach to operator attaches objects in one object set to objects in a different object set. The Attach to Self operator adjusts objects within the same object set to each other.

**Example**

Consider a map that shows two drawings.

One drawing, called **Lines**, contains a line while the other drawing, called **Areas** contains an area. The location precision has been set to less than the distance between the line and the area.

If we run the transform toolbar using [All objects in Lines] Attach to [All objects in Areas] then the line in **Lines** will be altered to become adjacent to the area in **Areas**.

The Attach to operator works with all types of objects. For example, areas may be attached to other areas.
Suppose we have a drawing called Gray with an area and another drawing called Yellow with another area.

If we used [All objects in Gray] Attach to [All objects in Yellow] the gray area would be adjusted to become adjacent to the yellow area.

Example

The Attach to transform will also move coordinates in parts of objects that are not near the target if the location precision distance is great enough.

Consider a drawing with three area objects, colored differently using thematic formatting to specify a different color for each area.
We will select the upper two objects and create a saved selection called Green.

We select the lower object and create a saved selection called Blue.

We run the transform toolbar with a setting of Green - Attach to - Blue and the areas in the Green saved selection will be attached to the blue object, but if the location precision distance is large enough (as it has been set for the example above) they will also be attached to each other and their other coordinates will be moved as well.

Example

Consider a drawing with three area objects, colored differently using thematic formatting to specify a different color for each area.
We will run Attach to Self using an object set of [All Objects] using three different settings for location precision. In each case we will begin with the drawing above, specify the location precision distance and then run Attach to Self.

As we increase the value of location precision distance (that is, making operations less precise)...  

...coordinates will be moved more aggressively because more of them fall within the location precision distance of each other.
When working with Attach to or Attach to Self the setting used for location precision will be critical, as this value must be set appropriately given the nearness of objects being attached to objects to which they are to be attached.

Although it is theoretically possible to attach objects to themselves using the Attach to operator, if this is what we have in mind we should use the Attach to Self operator which in most cases produces much better results.

See Also

View - Properties - Precision
**Transform - Boundaries**

The **Boundaries** operator in the Transform toolbar for drawings will add line objects that exactly coincide with the periphery of every area object in the target set.

In the illustration above, boundary lines have been created for the four Western US states areas. The boundary lines have been selected and moved slightly in the illustration at right to show clearly the addition of lines.

Note that creating a boundary line around each area will create coincident lines along places where the periphery of one area coincides with the periphery of an adjacent area. This may be seen in the above example by dragging apart each of the boundary line objects.

**Tech Tip**

Many transforms automatically run a Normalize Topology transform before running to eliminate common errors that may affect correct operation. The **Boundaries**, Points and Node Points transform operators do not normalize target objects before running. This makes it possible, for example, to segmentize lines with a length threshold of 10 meters, and then use the **Points** transform to place point objects at the locations of resulting inflection points.

The **Boundaries** operator will not split resulting objects if the **Split branches objects after transforms** option is turned off in the Tools - Options dialog.

**See Also**

Transform - Bounded Areas to create area objects from boundary lines.
**Transform - Bounded Areas**

The **Bounded Areas** operator in the Transform toolbar for drawings will add an area object in every region that is enclosed by lines, including those regions created by lines that intersect themselves.

In the illustrations above, four new areas have been created.

**Holes and Islands**

The **Bounded Areas** transform operator will try to resolve regions enclosed by lines within other closed border lines as separate areas. This is a very useful capability when creating lines from closed contours.

Suppose we have one region enclosed by lines that is entirely within another region enclosed by lines. We can select the lines and run the **Bounded Areas** transform operator from the transform toolbar.

By using **Touch Select** and touching both areas in turn we can see that the result is two areas have been created. One area is the inner circle and the other area is the larger circle with a "hole" in it where the inner circle area is located.

**Creating Areas in Drawings**

The **Bounded Areas** operator is frequently used when we import a data set that consists of lines and we would like to have areas. For example, we might import a drawing that outlines lakes or other regions using lines. We can use such drawings to create areas for lakes.
Suppose we've imported a drawing showing Great Bear Lake in Canada drawn as a boundary line. We can select the lines and run the **Bounded Areas** transform operator. The drawing is shown with **Background** turned off in the Layers pane so that the default checkerboard pattern shows through and makes it clear that only lines are involved.

The result is areas created within enclosed lines.

Zooming far into the region in the middle Eastern part of the lake we can deselect the largest area in the lake to see that smaller enclosed regions (representing islands) were created as separate areas by the **Bounded Areas** operator.
If we wished to create a single area for the lake to use within a map we might not want the other areas and can easily (since they are selected) delete them.

**Notes**

Be careful (and skeptical) when working with drawings that have been imported from CAD systems and other topologically unaware software. Such software might create lines that appear to be closed to form a region, but that upon examination at very high zoom levels will be seen to not connect to form closed regions. If the operator yields unexpected results, examine the lines that were thought to be closed for small breaks. For example, if selected contour lines do not result in neat areas it may be possible that the contour lines are not closed figures.

If the lines involved have data attributes, the created areas will acquire the data in accordance with the transfer rules that have been specified for those data columns.

**See Also**

Transform - Boundaries to create boundary lines from area objects.
Transform - Bounding Boxes

Create an area in the form of a rectangular bounding box that encloses each object and is aligned vertically and horizontally with the map window.

In the illustration above we have created a bounding box for the province of Zacatecas in Mexico.

Bounding boxes are created as areas and are selected when they are created. We would normally move them to their own layer so that they can be formatted and placed where desired in the layer stack. If we would like to have bounding boxes as lines we can take the areas and use the Boundaries transform toolbar operator to create lines.

The Common Bounding Box transform operator is like Bounding Boxes, but creates one bounding box about all objects in the scope.

In the example above the green areas were created with Bounding Boxes using the lines and the blue area was created with Common Bounding Box.
**Transform - Buffers**

Create a buffer zone for each object extending outward by the given distance and including the original object. For areas, this is an outer buffer zone.

Units used for the distance value in the source / argument box are taken from the projection of the drawing. If the drawing is in *Latitude / Longitude* projection the units will be degrees. If the drawing is projected, the units will be in meters or feet. It is therefore strongly recommended that the drawing be projected first so that meters or other linear measure can be used instead of attempting to use degrees as a measurement unit.

To see which units are defined for the drawing's projection, click on the Tracker tool and make some measurements. Whatever units are used to show the result of the tracker tool are the units currently in force. The tracker tool is also useful for estimating the desired size range of the buffer zone.

If the drawing uses degrees and meter-based creation of buffer zones are desired, first re-project the drawing into any convenient meter-based projection (such as *Orthographic*) so that meters will be used as the units of measure.

See the Transform - Border Buffers topic for a detailed example illustrating the creation of buffer zones.

**Buffer Zone Variations**

The transform toolbar provides four different buffer zone variations.

Variations are shown based on the area shown above in blue dotted style. All variations were created by showing the resultant buffer zone area as a 50% transparent layer in a map.
The **Buffers** operator creates an outer buffer zone, extending 15000 meters outward from the boundary of the area and including the original area.

**Inner buffer zone**, comprising all parts of the original area except those extending 15000 meters inward from the boundary of the area.

**Border buffer zone**, extending both inward 15000 and outward 15000 from the boundary of the area.

The **Common Buffer** transform operator is like **Buffers**, but creates one buffer zone about all objects in the scope.

In the example above the green areas were created with **Buffers** using the lines and the blue area was created with **Common Buffer**. Note that there are three area objects in the case of the green buffer zones but only a single area object in the case of the common buffer.

**Local Scale and Units of Measure**
Stating that the units used by buffer operators are the units used in the projection is actually an oversimplification. More accurately, the parameter value is taken as the number of local drawing units. Local drawing units are normally the same as the units used by the projection. Local drawing units are the projection units multiplied by the **Local scale** value for the coordinate system specified in the Edit - Change Projection dialog.

Because the **Local scale** value is usually 1, the value used in the source / argument box for buffer operators is usually the same number in local drawing units. Entering 1500 in a meter-based projection thus means 1500 meters. However, if the **Local scale** value is not 1, the value in the source / argument box must be multiplied by the **Local scale** value to get the number of units that will be used to create the buffer. If the **Local scale** value is 10 in a meter-based projection, for example, then entering 50 into the source / argument box is not a command to create a 50 meter buffer, it is a command to create a 500 meter buffer.

An additional subtlety must be considered when this transform operator is used with a drawing layer within a map. If the map component's projection is degree based while the native projection for the drawing is meter based (as, for example, with the **Orthographic** projection) or vice versa, then the transform operator will use whatever units are used by the drawing's projection.

**Tech Tip**

Variations on the basic buffer zone constructions with areas may be easily created by first creating a buffer zone and then using **Clip with (Intersect)** or **Clip with (Subtract)** between the new buffer zone area object and the original area object. For example, using a **Clip with (Intersect)** between a border buffer and the original area will produce a zone that consists only of the defined distance within the area's boundary and that excludes the more inner regions of the original area as well as everything outside of the boundary.

The buffer distance used must be at least four times the size of the location precision factor. If precision has been set to 10 meters then the buffer distance must be larger than 40 meters.

**See Also:**

**Transform - Inner Buffers**
Transform - Centroids

Three related transform operators create centroids for objects using different methods. Centroids are points placed at the "center" of an object. They are used for many reasons but perhaps the most important is to simplify and abstract geographic data in the form of areas or lines into the much simpler form of points. At times we will want the data in a simpler form for export to other programs or to use analytic methods that work with points but which do not work with areas or lines.

Centroids | Create a point at the center of the minimum circle that encloses each object. (areas or lines)
Centroids (Box) | Create a point at the center of the minimum enclosing rectangle for each object. (areas or lines)
Centroids (Inner) | Create a point at the "center" of an area and adjust the position of the point so that it always falls within the area. (areas only)
Centroids (Weight) | Create a point at the approximate center of balance of each object. (areas only)

Centroid operators do nothing for points, since a point is always its own centroid. Two of the three operators can work with lines as well as with areas. When a centroid object is created it inherits the data fields of the object from which it was created. The illustrations above show centroids created for areas with the Centroids operator.

The Common Centroid transform operator is like Centroids, but creates one centroid using the center of the minimum enclosing circle for all objects in the scope.

Examples

In the examples that follow we will use the province of Zacatecas in Mexico as a sample area.

The Centroids (Weight) operator creates a point at the approximate center of balance of the area. It uses a fast algorithm that will usually, but not always, place the centroid point within the area. Very strange area shapes such as horseshoe shapes will cause the centroid point to be placed outside the area.
The **Centroids** operator draws a minimum enclosing circle about each area and creates the centroid at the center of the circle. The illustration above shows the circle centroid in red. Note that the position of the circle centroid is different from the centroid computed for the approximate center of balance.

The **Centroids (Box)** operator draws a bounding box about each area and creates the centroid at the center of the bounding box. The illustration shows a bounding box superimposed above an enclosing circle with the box centroid shown as a small square dot.

**Centroids (Inner)**

We will often encounter areas where the centroid computed using the **Centroids**, **Centroids (Box)** or by the **Centroids (Weight)** transforms will be placed outside an area.

Consider a map of the Southeastern United States.
If we create centroids (green dots) using the Centroids transform we see that the centroid for Florida falls outside of the state. If we were too zoom far into the drawing we would see that the centroid created for Louisiana also falls outside that state.

We can use the Centroids (Inner) transform to create centroids (yellow squares) that are guaranteed to fall within their areas.

**Centroids and Lines**

Because an enclosing circle or a bounding box can be found for lines as well as for areas we can create centroids for lines using the box and circle centroid operators.

This illustration shows four lines with their centroids, computed using Centroids.
If we draw enclosing circles about each line (shown in red selection color) we can see how the locations of the centroids were determined.

Centroids are most frequently created for areas. However, they are also a useful means of "converting" line objects into point data. For example, the above illustration shows lines in a hydrography layer where what appear to be continuous lines are in fact many lines that abut one another.

Using the Centroids operator we can create centroids for each individual line.

As an example of why we would want to do so, suppose for each line segment we have the length of the line. We would like to know the total length of waterways per square kilometer in various regions of the map. We can approximate this by creating a grid where each box is one kilometer square and then creating centroids for each line segment. It is then an easy matter to add up the total "length" values for each centroid point that happens to be in each square kilometer grid box.

In the example above the green points were created with Centroids using the lines and the blue point was created with Common Centroid.

Tech Tip

Many transforms automatically run a Normalize Topology transform before running to eliminate common errors that may affect correct operation. The Centroid Circle, Centroid Box and Common Centroid transform operators do not normalize target objects before running.
Transform - Decompose

Split branched objects into simple (non-branched) objects. For example, a single, branched area object composed of three "islands" will be split into three separate area objects.

Example

Many drawings will show a mainland region and islands as composed of the same area object. If we wish to delete some of the islands we might first want to use decompose to convert the single, complex, branched area object into many unbranched objects. We can then delete the unbranched objects quickly.

Suppose we have such a drawing. If we click with select touch on any of the "areas" we see they are all selected.

All of the "areas" are selected at once because they are all part of the same area object. We can use decompose to split them apart.

To do so, we load up the transform toolbar as seen above and push Apply.

Thereafter, if we click with select touch on any of the areas we see that only the object touched is selected. This is because all of the "objects" we see are now truly separate area objects.

Note: some GIS systems may refer to this function as "explode."
Transform - Decompose to Convex Parts

The **Decompose to Convex Parts** operator splits area objects into smaller areas automatically. Each smaller area created will be a convex hull for the locations it encloses. This operator is often used to split area objects into smaller areas to allow computations or to create centroids that fall within areas.

![Decompose to Convex Parts Example](image)

Given an area at left the **Decompose to Convex Parts** transform will split it into smaller areas as seen at right. Each smaller area is a convex hull for locations it contains. There are many possible decompositions of a larger area into smaller areas in this way. The algorithm generates one such decomposition.

Note that since each smaller area is a convex hull for the locations it contains, the centroid of each of the smaller areas is guaranteed to be located within the area.

![Centroid Example](image)

With some areas, as seen above, the centroid lies outside of the area. If centroids are created to allow transfer of areas to points via the Spatial Overlay dialog (that is, using "point in polygon" operations) for analytic or other purposes, the centroid point might not fall within the desired area. One approach to solving this problem is to split each area into smaller areas and to then use centroids for those smaller areas.

![Decomposed Areas](image)

We can use **Decompose to Convex Parts** to split the area into smaller areas.
The centroids for each will fall within the area. We can now transfer data using spatial overlays knowing that the fields for each area will transfer to a centroid point within that area.

**Notes**

- Use the Centroids (Weight) operator to create centroids that are more centered on each smaller area.
- The transfer of data fields to the new areas will be governed by the Transfer Rules for the fields.
Transform - Decompose to Triangles

The **Decompose to Triangles** operator automatically splits area objects into triangles. The triangles are created using the Delaunay criterion, which seeks to maximize the minimum angle of each triangle. This operator is often used to split area objects into triangles to allow computations or to create centroids that fall within areas. See the discussion of such usage in the Decompose to Convex Parts topic.

Suppose we have an area that we wish to decompose into triangles.

After running the operator the area will be replaced with two triangular areas.

The operator can, of course, be used with considerably more complex areas, such as those representing geographic entities such as the island shown above, located off the coast of Mexico.
Decomposing it into triangles results in many triangles. The triangles are created using the coordinates that define the area as vertices for the triangles.

Notes

- The transfer of data fields to the new triangular areas will be governed by the Transfer Rules for the fields.

See Also

- Triangulation
- Decompose to Convex Parts
The **Distance Network** transform operator creates a network link between all points within the given distance.

In the illustration above we've created a grid of points using the View - Grid tool. The points are 48 meters apart horizontally and vertically.

Setting up the transform toolbar as seen above with a distance of **50** will result in the following network.

Links have been drawn between neighbors in a vertical and horizontal pattern. The diagonals have not been drawn because the hypotenuse between points is greater than 50 meters.

If we delete the selected lines added by the transform and run the **Distance Network** operator one more time with a value of **70** we get the following result:

In this case links have been drawn on the hypotenuse diagonals as well because each point is 69 meters away from the diagonal neighbor (as is easily seen with the Tracker tool).

Note that in both of the above cases, the **Distance Network** operator created **two** links between each point: one in each direction. The extra links can be deleted by selecting all links and then running the Normalize Topology transform operator. This will delete duplicate coincident links.
Units of Measurement

Units used for the distance value in the source / argument box are taken from the projection of the drawing. If the drawing is in Latitude / Longitude projection the units will be degrees. If the drawing is projected, the units will be in meters or feet. It is therefore strongly recommended that the drawing be projected first so that meters or other linear measure can be used instead of attempting to use degrees as a measurement unit.

To see which units are defined for the drawing’s projection, click on the Tracker tool and make some measurements. Whatever units are used to show the result of the tracker tool are the units currently in force.

If the drawing uses degrees and meter-based creation of buffer zones are desired, first re-project the drawing into any convenient meter-based projection (such as Orthographic) so that meters will be used as the units of measure.

When this transform operator is used with a drawing layer within a map the units used by the drawing will be used. If the map component’s projection is degree based while the native projection for the drawing is meter based (as, for example, with the Orthographic projection) or vice versa, then the transform operator will use whatever units are used by the drawing’s projection.
Transform - Enclosing Circles
Create an area in the form of the minimum enclosing circle for each object.

In the illustration above we have created an enclosing circle for the province of Zacatecas in Mexico.

Enclosing circles are created as areas and are selected when they are created. We would normally move them to their own layer so that they can be formatted and placed where desired in the layer stack. If we would like to have enclosing circles as lines we can take the areas and use the Boundaries transform toolbar operator to create lines.

The Common Enclosing Circle transform operator is like Enclosing Circles, but creates one minimum enclosing circle for all objects in the scope.

In the example above the green areas were created with Enclosing Circles using the lines and the blue area was created with Common Enclosing Circle.

Tech Tip

Many transforms automatically run a Normalize Topology transform before running to eliminate common errors that may affect correct operation. The Enclosing Circles transform operator does not normalize target objects before running.
Transform - Enclosing Rectangles
Create an area in the form of the minimum enclosing rectangle for each object, allowing the rectangle to be rotated as compared to the map window.

Given the two lines above, applying the Enclosing Rectangle operator creates the areas shown below. Note that since there are two lines, a minimum enclosing rectangle is created for each.

The Common Enclosing Rectangles transform operator is like Enclosing Rectangles, but creates one minimum enclosing rectangle for all objects in the scope.

In the example above the green areas were created with Enclosing Rectangles using the lines and the blue area was created with Common Enclosing Rectangle.

Tech Tip
Many transforms automatically run a Normalize Topology transform before running to eliminate common errors that may affect correct operation. The Enclosing Rectangles transform operator does not normalize target objects before running.
**Transform - Explode**

Explode lines into multiple line objects where each line segment from the original line becomes a separate line object. This transform operator has no effect on areas or points.

When new lines are created their data attributes will be taken from the original line according to the transfer rules specified for the table's columns.

**Example**

Consider a drawing with one line, that has been selected.

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>Hwy 101</td>
</tr>
</tbody>
</table>

In the drawing's table we can see there is only one object. The **N to 1** transfer rule for the **Name** column has been set to **Copy**.

If we run the **Explode** transform, we convert each segment of the line into a separate line object. We can select a few segments to show they are now separate lines.

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>Hwy 101</td>
</tr>
<tr>
<td>18</td>
<td>Hwy 101</td>
</tr>
<tr>
<td>19</td>
<td>Hwy 101</td>
</tr>
<tr>
<td>20</td>
<td>Hwy 101</td>
</tr>
<tr>
<td>21</td>
<td>Hwy 101</td>
</tr>
<tr>
<td>22</td>
<td>Hwy 101</td>
</tr>
<tr>
<td>23</td>
<td>Hwy 101</td>
</tr>
</tbody>
</table>

The drawing's table now shows clearly that there are multiple lines. The effect of the **Copy** transfer rule is to copy the value of the **Name** column into each of the new line objects created.
Note also that there is no longer an object with an ID of 16 because the original line object has been deleted. It has been replaced by the newly created lines, one for each of the former segments.
Transform - Join Lines

Join lines with coinciding ends into a new, single line object. Redistribute any data attribute fields according to transfer rules.

In the example shown above, suppose A1, A2, A3, B1 and B2 are all separate line objects. Suppose further that the ends of the lines are exactly coincident and do not have gaps between them as shown in the illustration

Running the Join Lines transform on the selection contain these five line objects will create two new line objects.

One new line object will be made up of A1 + B1 + A2. The other new line object will be made up of A3 + B2.

Technical Note

Given an arbitrary set of lines it can be remarkably difficult to join "coincident" lines. After all, consider that lines can be multi-branched objects like areas. In the case of multi-branched lines what appear to be two different line objects can be the same object. For example, in the illustration above a topologically complex (and weird) drawing could have B1 and B2 both be branches of the same line object and A1, A2 and A3 also be three different branches of the same line object.

Given a set of lines the Join Lines transform creates a network with a link for each branch of each line and then composes another set of lines that collapse chains within the network. Note that chains could consist of parts of different objects). This algorithmic approach combining graph theory and computational geometry provides a flexible, reliable transform operator that can deal with arbitrarily complex topology.

See Also

Dissolve - The Dissolve tool can join lines by data attribute. It uses the Join Lines algorithm to join lines together that have the given data attribute.
T
**Transform - Intersection Points**

The **Intersection Points** operator in the Transform toolbar for drawings will add a point object at every location where lines intersect each other, including self-intersections.

This command does not change the lines in any way. It simply adds points at the places where lines cross each other.

**See also:**

- Transform - Intersect Lines to add points at intersections and also split lines into separate line objects at intersections.
- Transform - Points to add points at all coordinates defining the shape of lines and areas.
Transform - Node Points

The Node Points operator for drawings creates a point object at the end of every line object. If the ends of several line objects coincide, only one node will be created at that location. This transform is used to build true networks out of systems of lines.

Suppose we have three lines that are incident to each other at a location. If we would like to treat these three lines as a network we need to create a point at the ends of the lines.

Applying Node Points creates a point at the end of each line. Only one point is created at the location where all three lines are incident to each other.

Note the difference between the Node Points and the Points transform.

Had we run the Points transform on the initial drawing we would have obtained the result above. The Points transform adds points at every coordinate location that defines the shape of the lines. The Node Points solver adds a point at only the terminal coordinate locations.

Nomenclature

Manifold can treat any system of lines and points as a network. When we discuss network subjects within Manifold, we shift terminology so that points are called nodes and lines are called links. People unfamiliar with networks will often refer to a system of lines such as that in the first illustration above as a "network." However, a network is defined by its nodes, not by its links. Links are simply a way of describing how the nodes of a network are formed into a network. If a drawing (or map) has no points, it is not a network.

Some GIS systems inaccurately use the word node to mean one of the coordinate locations that defines the shape of a polyline. That's a grave error since the world of networking (a considerably larger user population than GIS) in both computers and mathematics uses the word node to mean a point that is a vertex in a network.

Since GIS users will often wish to use a system of lines as a network even if it contains no nodes, Manifold takes a less strict view: default operators in Manifold will treat as a network any system of lines where the endpoints of the lines exactly coincide even if no points exist at the ends of the lines. Use the Node Points transform to add points to create a true network. This will avoid trouble with other software or scripts that may apply sophisticated network mathematics and insist that the "network" in use is a true network.

Tech Tip
Many transforms automatically run a Normalize Topology transform before running to eliminate common errors that may affect correct operation. The Boundaries, Points and Node Points transform operators do not normalize target objects before running. This makes it possible, for example, to segmentize lines with a length threshold of 10 meters, and then use the Points transform to place point objects at the locations of resulting inflection points.

See Also

About Networks
**Transform - Points**

The Points operator in the Transform toolbar for drawings will add a point object at every coordinate location defining the shape of objects in the target box.

![Diagram](image)

When applied to lines, note how a point is created at each of the coordinate locations where the line changes direction.

This command is often used to "convert" areas or lines into sets of points in the shape of those objects.

**Tech Tip**

Many transforms automatically run a Normalize Topology transform before running to eliminate common errors that may affect correct operation. The Boundaries, Points and Node Points transform operators do not normalize target objects before running. This makes it possible, for example, to segmentize lines with a length threshold of 10 meters, and then use the Points transform to place point objects at the locations of resulting inflection points.

**See Also**

Transform - Intersection Points to add points to locations where lines intersect.

Transform - Node Points to add points to systems of lines so that the resulting drawing or map can be used as a true network.
Transform - Clip with (Intersect) / (Subtract)

The Clip with operators in the Transform toolbar for drawings use one or more areas to "clip" other objects, leaving only those parts of the other objects that lie within the areas (Intersect) or removing those parts of the other objects that lie inside the areas (Subtract).

For clarity, the following examples show use of only one clipping area. However, more than one area may be in the "clip with" set. The examples use one area to clip lines. The commands also work when areas are used to clip other areas, no matter how topologically complex the areas involved may be.

Clip with (Intersect)

Using a drawing that shows the state of Texas overlaid with lines forming a grid, we first select the state of Texas and save it within the View - Selections dialog as a saved selection called Texas. We can select the lines and then apply the Clip with (Intersect) command in the Transform toolbar:

The Transform toolbar knows we have made a selection, so the target box is loaded with the selection as the target. We choose Texas in the source / argument box (all of our saved selections will appear as possible choices in the source / argument box) and press Apply.

The result is to split all objects at their point of intersection with the periphery of the Texas area and to discard all objects lying outside of Texas. This is called an "intersection" because it retains all objects where there is an object and there is Texas.

Suppose we have two drawings in a map. Drawing B contains green rectangles and Drawing A contains blue circles (seen with 50% opacity above the rectangles layer).
If we use [All Objects in Drawing B] Clip with [Intersect] [All Objects in Drawing A] in the transform toolbar the result is seen above. Drawing A has been turned off. What is left of the rectangles is the intersection between objects in Drawings A and B. Note that this intersection operation removed everything from the rectangles except the region of intersection. If we wanted to retain the original rectangles as well as the intersections we could have made a copy of Drawing B and used that in the Clip with [Intersect] operation.

Clip with (Subtract)

Beginning once again with Texas overlaid with a grid of lines, we can apply the Clip with (Subtract) command.

The toolbar is the same, except we choose Clip with (Subtract) and hit Apply.

The result is to split all objects at the boundary of Texas and to subtract all objects that lie within Texas.

Clip with Works with All Objects

These commands also work when areas are used to clip other areas, no matter how topologically complex the areas involved may be.
The illustrations above show Clip with (Intersect) and then Clip with (Subtract) applied to two areas using Texas as the "clip with" area.

It is often the case that we have a map showing roads, lakes and other objects throughout an entire region and we wish to make a map that shows only the objects within a part of the region, such as a specific state or province. The Clip with commands are a straightforward means of cutting out only those objects within the desired state or province. Simply use the state or province as an area object with Clip with (Intersect) to cut out only those parts of the objects that are within the desired area.

To keep the illustrations in this topic simple, only illustrations using either lines or areas have been used; however, the commands work perfectly no matter what mix of objects is in the target set.
The transform toolbar includes two operators to find clusters: the Clusters and the Clusters (Zahn) operators. The two operators both find clusters of points but use slightly different approaches. Both employ manifold.net graph theoretic algorithms to find clusters.

- The Clusters form uses relative neighborhood networks to distinguish clusters based on a combination of network and geometric relationships. Applying this operator with a parameter of 0 results in a relative neighborhood network. Relative neighborhood networks incorporate reckoning of the overall arrangement of a point set and are superior to ordinary, local nearest neighborhood methods.

- The Clusters (Zahn) form uses minimum spanning tree networks to distinguish clusters based on minimum path spanning trees. It is named to honor Charles T. Zahn for his work in describing the general use of minimum spanning tree graphs in cluster detection. The links created using this operator are a subset of the links created when a minimum spanning tree is formed. Minimum spanning trees incorporate links based on the minimum traversal of a branched tree that reaches all points, so this form may better reveal clusters arising from spatial propagation.

Clusters are simply collections of points that appear to form groups or to otherwise be related to each other. We may use various mathematical techniques to identify them; however, what counts in most GIS analysis and data mining is the use of cluster-finding methods to identify groups of points for further examination. The essential thing is if the software reveals patterns that make sense to us once we see them but which otherwise could not be found by eye.

The Clusters operators work by creating links between points that are parts of a cluster. The value given in the source / argument box guides the operation. Smaller parameters will result in larger clusters.

Suppose we begin with the set of points shown above. If we apply the Clusters operator with a parameter of 50 it will create lines between points for the clusters found as shown below:
By increasing or decreasing the parameter we can force fewer or more of the unconnected points to be assigned to a cluster.

The (Zahn) form of the operator results in the above clusters when run with a parameter of 50.

**See Also**

Relative Neighborhood Network - The transform operator that creates relative neighborhood networks. This topic includes a note on characteristics of relative neighborhood networks.

**Spanning Tree** - The transform operator that creates minimum spanning trees.

**Notes**

In graph theory, networks are called **graphs**. When searching Internet for information on these topics try searching for words like "relative neighborhood", "graph", "spanning tree", "cluster" and similar. These ideas are applied in an astonishing range of disciplines, from fungal spore distribution patterns to the characterization of finds in archeological sites.
The points for the examples above were created by making centroids for provinces in Mexico using the **Centroids (Weight)** transform operator. The points and results of the transform operators are shown in a map with a drawing of Mexico as a backdrop. We used the centroids as a source set of points because they are dispersed in a geographically interesting way. There is no meaning to the map of Mexico; however, if we were to place one antenna in the geographic center of each province and then we wished to link the antennas in a network we would likely create and study maps such as these.
Transform - Constrained Triangulation

Similar to the triangulation transform, the constrained triangulation transform creates a constrained Delaunay triangulation of a point set by treating the points as nodes in a network and drawing links between them that divides the region between the points into triangular tiles.

There are two differences between triangulation and constrained triangulation:

- Unlike the triangulation transform, the constrained triangulation transform will not draw links through pre-existing line segments.
- The triangulation transform operates only on point objects. The constrained triangulation transform can operate on any object, and will treat the coordinates of lines and areas (inflection points) as points for the triangulation.

There are three transform operators for constrained triangulation:

- The Constrained Triangulation Lines operator creates lines for the triangulation.
- The Constrained Triangulation Areas operator creates areas for the triangulation.
- The Constrained Triangulation operator creates both lines and areas for the triangulation.

See the Transform - Triangulation topic for a general discussion of triangulation.

Example

If we take a set of points as shown above and apply the Triangulation Lines transform operator...

...we can create a triangulation consisting of lines, shown in blue above.
Suppose, however, if we had started with both points and lines (shown in yellow) as seen above and applied the "Triangulation Lines" transform operator.

The result would be the same triangulation, with some of the lines created for the triangulation crossing the pre-existing yellow lines. There are many applications where we would not want a triangulation crossing pre-existing lines. For example, perhaps such lines mark the boundaries between contours or show highways or property boundaries which the triangulation should not cut. We can respect pre-existing lines by using constrained triangulation.

Suppose we start with a set of points and lines and then apply the "Constrained Triangulation Lines" transform operator.
The result shown in light violet color shows that none of the triangulation lines cross any of the pre-existing lines.

We can see the difference between the lines created by triangulation and those created by constrained triangulation by showing the two together as layers in a map with constrained triangulation lines shown in light violet color and triangulation lines shown in thinner blue lines.

The red lines above show lines created in triangulation that are not created in the constrained triangulation.
Constrained triangulation creates a new line, shown in red above, that is not created by regular triangulation.

**Example**

Unlike regular triangulation, which only works with points, constrained triangulation can work with points, lines and areas. The coordinates of lines and areas are taken as points for the purpose of constrained triangulation. This allows us to create constrained triangulations using lines or areas.

The classic example might be creating a triangulation on a set of contour lines. If we begin with the lines seen above and apply the **Constrained Triangulation Areas** operator...
...we create the triangulation seen above, where the triangulation areas are colored in gray. Note that because of the constrained nature of the triangulation no created area cuts across a pre-existing contour line.

**Tech Tip**

Depending on the location of pre-existing segments, the triangulation created by the transform may or may not conform to the Delaunay principle.

**See Also**

- Decompose to Triangles
- Transform - Triangulation

Other types of networks easily created with transform operators:

- Gabriel Network
- Relative Neighborhood Network
- Spanning Tree
**Transform - Convex Hull**

Given a set of points, this operator creates an area about the points using the positions of the extremal points as the bounding coordinates for the area. This is easiest to visualize if we imagine the points to be vertical pegs set on a board and a rubber band is placed around the collection of points to define the shape of the area.

![Diagram of Convex Hull](image)

Running the convex hull transform on the points at left results in the area seen at right. The area line created by this operator will be selected after it is created.

**Tech Tip**

Many transforms automatically run a Normalize Topology transform before running to eliminate common errors that may affect correct operation. The Convex Hull transform operator does not normalize target objects before running.
**Transform - Flip Horizontally**

Flip objects left / right to their mirror image. Because calculations must be made to find the center of the objects to be flipped, it is strongly recommended that drawings be projected before using this operator.

**Example**

We begin with a map of Europe in **Orthographic** projection. The area representing Italy has been selected. Next, load the Transform toolbar with

```
[selection] Flip Horizontally
```

and press **Apply**. This is a command to flip Italy left / right to its mirror image.

This command is most frequently used in CAD-style editing of non-geographic data. It is shown here in a geographic setting to make clear its effect.
Transform - Flip Vertically

Flip objects top / bottom to their upside down image. Because calculations must be made to find the center of the objects to be flipped, it is strongly recommended that drawings be projected before using this operator.

Example

We begin with a map of Europe in Orthographic projection. The area representing Italy has been selected. Next, load the Transform toolbar with

[selection] Flip Vertically

and press Apply. This is a command to flip Italy left / right to its mirror image.

This command is most frequently used in CAD-style editing of non-geographic data. It is shown here in a geographic setting to make clear its effect.
Transform - Gabriel Network

Given a set of points this operator builds a Gabriel network on that point set by creating new lines to serve as the links. **Note:** In a classic Gabriel network, the set of points should not include any co-incident points, that is two points which lie exactly at the same location.

A **Gabriel network** is a network where the links are determined by making a pairwise comparison of points in the context of the points around them. We start with a set of points, and consider all possible pairs of points in the set. For each pair of points, we draw a link between them if there are no other points which lie inside a circle whose diameter is the distance between the points and which is defined by the two points.

The construction above shows a Gabriel network for a set of five points. Note that if any other pairs of points are considered, the circle defined by them would include at least one other point and thus the link is not drawn.

The lines created by this operator will be selected after they are created. It's a good idea to move them to a new drawing to keep the map well organized.

**Example**

Using the set of points above, we can create the Gabriel network seen below.
Note that a subset of a Gabriel network is created by the Relative Neighborhood Network transform operator as seen below:

The Gabriel network is a good way to organize a set of points into a network that maintains a reasonable overall shape of the point set. The relative neighborhood network refines this to create a sparser network that takes greater account of each local point clustering compared to surrounding points. An even sparser network can be created using the Minimum Spanning Tree operator, the results of which are seen below:
The minimum spanning tree removes loops that are present in the relative neighborhood network and creates a network that uses the minimum total length of links between points that connects all of the points.

**See Also**

Relative Neighborhood Network - The transform operator that creates relative neighborhood networks, which are subsets of Gabriel networks. This topic includes a note on characteristics of relative neighborhood networks.

Spanning Tree - The transform operator that creates minimum spanning trees.

Clusters - Transform operators that use relative neighborhood networks and minimum spanning trees to find clusters.

**Notes**

In graph theory, networks are called graphs. When searching Internet for information on these topics try searching for words like "gabriel graph", "relative neighborhood", "graph", "spanning tree", "cluster" and similar. These ideas are applied in an astonishing range of disciplines, from fungal spore distribution patterns to the characterization of finds in archeological sites.

The points for the examples above were created by making centroids for provinces in Mexico using the Centroids (Weight) solver. The points and results of the transform operators are shown in a map with a drawing of Mexico as a backdrop. We used the centroids as a source set of points because they are dispersed in a geographically interesting way. There is no meaning to the map of Mexico; however, if we were to place one antenna in the geographic center of each province and then we wished to link the antennas in a network we would likely create and study maps such as these.
**Transform - Intersect Lines**
Splits lines into multiple line objects at all places where lines intersect.

If two lines intersect, they will be split into four line objects, as shown above.

The second example shows one long line in a spiral shape that is intersected in several places by a diagonal straight line. **Intersect Lines** will split the two lines into fourteen separate lines at the intersection locations marked with red plus characters.

See also: Transform - Intersection Points to add points to locations where lines intersect without altering the lines.
Transform - Move Horizontally / Move Vertically

Move all objects or selected objects the specified distance horizontally or vertically. Negative values move objects to the left (West) or down (South).

Units used for the distance to be moved value in the source / argument box are taken from the projection of the drawing. If the drawing is in Latitude / Longitude projection the units will be degrees. If the drawing is projected, the units will be in meters or feet. It is therefore strongly recommended that the drawing be projected first so that meters or other linear measure can be used instead of attempting to use degrees as a measurement unit.

If the drawing uses degrees and meter-based moves are desired, first re-project the drawing into any convenient meter-based projection (such as Orthographic) so that meters will be used as the units of measure.

Example

We will move Germany in a drawing of Europe. The drawing has been projected into Orthographic projection, so it is a meter-based drawing.

We first select the area representing Germany. Next, load the Transform toolbar with

[selection] Move Horizontally -200000

and press Apply. This is a command to move the area 200 km to the West.

To show vertical movement we load the transform toolbar with

[selection] Move Vertically -300000

and press Apply. This is a command to move the area 300 km to the South.
While it is something of an absurd example to move countries this way, these commands are well-suited to adjusting data sets, such as drawings of wells or other points, that contain systematic errors. They are also useful when using Manifold for CAD editing.

**Tech Tip**

Many transforms automatically run a Normalize Topology transform before running to eliminate common errors that may affect correct operation. The Move transform operators do not normalize target objects before running.
Transform - Nearest / Farthest Neighbors

**Nearest Neighbors:** Given the source set objects in the drawing and a target set of objects in the source / argument box, for each point in the source set finds those points in the target set which are closer to that point than they are to other points in the source set. The neighbors are indicated by a line drawn between them. This is a geometric solver, using straight-line distances, not distances through a network.

This is a general-purpose client-center assignment solver. Suppose we have a number of service centers (the source set) and we also have a collection of customer points (the target set). We would like to assign each customer to the nearest service center. This solver does it automatically: for each service center it finds the "nearest neighbor" from the customer set. It guarantees that no reassignment of a customer to a different service center will reduce the distance between that customer and a service center.

The operator creates lines in the active drawing, which will be selected. It's probably a good idea in most cases to move the lines to a different drawing for better organization.

**Farthest Neighbors:** The same as above, except that for each point in the source set finds those points in the target set which are farther from that point than they are from any other point in the source set.

This solver is used when two classes of things must be kept far apart. Suppose we own a chain of 500 restaurants that are administered through six regional offices. We have a collection of auditors who work with computerized records only, and thus could be located anywhere; however, we have them located in the various regional offices because from time to time they pitch in with other work. We would like to assign each auditor a group of restaurants that are as far away from his region as possible, so that in the course of ordinary business he or she won't get too friendly with the restaurant managers they may be auditing.

Using the Farthest Neighbors solver will make this assignment. It guarantees that no reassignment of an audited restaurant to different auditing manager can increase the distance.

If the points involved have data attributes, the created lines will acquire the data in accordance with the transfer rules that have been specified for those data columns.
Transform - Normalize Metric

Normalize Metric repairs common flaws in object metrics in drawings. Objects in drawings are defined by their coordinates (see the Editing Objects topic). The sequence of coordinates that defines an object is called its metric.

The Normalize Metric operator will fix:

- Self-overlaps
- Self-intersections
- Redundant internal coordinates (duplicate coordinates, unnecessary coordinates)
- Duplicate Coordinates
- Lines with redundant point features in the metric.

Redundant linear features in areas are sequences of internal coordinates that jump from a given coordinate out to some other coordinate and then back to identically the same coordinate before continuing the sequence that defines the area object. A redundant point feature in a line is a repeated coordinate in the sequence defining the line.

Examples

The illustrations above show an area containing a redundant linear feature in the metric. The coordinate sequence defining the area jumps out from a particular coordinate and then back again to exactly the same coordinate. The "line" that appears to exist is a dimensionless phantom.

Normalize Metric will remove the two unnecessary coordinates (one outside the area and a redundant duplicate at the same location in the area boundary).

Optimization

Once Normalize Metric or Normalize Topology is run on an object, the object will be flagged within internal Manifold storage as clean. During future Normalize runs the process will skip examination and normalization of objects that are marked clean. If an object is subsequently modified (through editing by a user or by a script) the "clean" flag will be cleared so that the object will be examined by future passes of Normalize Metric. If the Location Precision property of a drawing is modified via the View - Properties - Precision dialog to a higher value (that is, a less demanding precision) then the "clean" flag for all objects will be cleared.

Because of the caching behavior of Normalize Metric, the amount of time required for the function will vary based on how many objects are been marked "clean." If the Location Precision has just been changed in a large drawing the function will take longer to run. If Normalize Metric has just been run (either explicitly or automatically as part of another transform), the function will run very rapidly since all objects will be marked "clean."
Automatic Normalization

Manifold maintains automatic cleaning of object metrics. Automatic normalization occurs for those transform operators that require clean object metrics. Before any such Transform toolbar operator begins operation, Normalize Metric automatically will be run on all objects that need to be cleaned.

The automatic normalization is an internal Manifold function that occurs whenever it is required during transform toolbar operation. Due to the caching behavior of Normalize Metric the amount of time for cleaning (if required) will vary depending on whether or not objects in the drawing are marked "clean" or not. A particular transform operator might run very rapidly and then just after Location Precision has been changed suddenly run much more slowly during the next run. The slower run occurs because all of the objects have had their "clean" flag cleared as a result of the precision change, so Normalize Metric will have more work to do for that run.

Differences between Normalize Topology and Normalize Metric

Both of these transform operators are used to produce a clean metric; however, the metric produced by Normalize Metric may be slightly different than that produced by Normalize Topology. The main difference between them is that Normalize Metric is run on a per-object basis while Normalize Topology is run on an object set as a whole.

When processing metrics Normalize Topology considers neighboring objects and Normalize Metric does not.

Examples:

- Normalize Topology will detect overlapping areas and will assign the region of overlap to one of the overlapping areas while clipping it from the others. In this same circumstance Normalize Metric will leave the overlaps.
- Normalize Topology will snap points to the nearest object (such as the end of a line) if the distance between the point and the object is less than the location precision. Normalize Metric will leave points unmoved.
- Normalize Topology will detect an intersection between two lines and will split the lines at the intersection. Normalize Metric will leave the lines unchanged.
- Normalize Topology will remove duplicate coordinates as well as other redundant coordinates. Normalize Metric will only remove duplicate coordinates.

Both Normalize Topology and Normalize Metric will remove duplicate coordinates, but Normalize Topology will also attempt to remove redundant coordinates. Normalize Topology will leave any redundant coordinates that are also coordinates in any other object. This is done to guarantee that a common border between any two adjacent objects is exactly the same in both objects. Note that Normalize Metric can not do the same since it operates only on single objects.

How should one choose between Normalize Topology and Normalize Metric? In most cases, we would like any adjustments to object metrics to proceed with the entire object set in mind so in most cases one would use Normalize Topology in preference to Normalize Metric.

When dealing with imported drawings that may contain specific features of the metric that are to be preserved, users will often choose to begin operations by running Normalize Topology. This will clean object metrics while preserving redundant coordinates that are co-located with other objects and it will mark all objects as having clean metrics. This avoids a situation where Manifold is compelled to run Normalize Metric even though we do not want it run.

If some other transform operator (such as Clip Intersect) is run on a drawing that has not been yet normalized with either Normalize function Manifold has no choice but to run Normalize Metric to guarantee a clean metric. By running Normalize Topology first we pre-empt the running of Normalize Metric by other transforms in self-defense. The main reason for running Normalize Topology first is to prevent the deletion of redundant coordinates that are co-located with other objects. A secondary (rare) reason is to prevent redundant coordinates from being deleted when we are working at a location precision that may cause coordinates to be judged "redundant" that are really needed to prevent gaps from emerging between formerly adjacent objects.

Why do the other transform operators run Normalize Metric instead of Normalize Topology if objects need cleaning before processing? Transform operators are frequently run on subsets of objects. If Normalize Topology were used automatically the usual workflow of running transforms on subsets of objects would quickly lead to a clustered drawing, different parts of which have been normalized with respect to different sets of objects. That would defeat the global normalization that is usually a key objective of Normalize Topology.
To summarize, when in doubt run **Normalize Topology** once before running other transforms. This can be very time consuming for large drawings, but it is a prudent and necessary step experts will take if they wish to preserve specific metrics.

**See Also**

**Transform - Normalize Topology**
Transform - Normalize Topology

Normalize Topology alters drawings in many ways at once by making fine adjustments in the positions and coordinate points defining points, lines and areas. If objects fall within the distance specified in Location Precision in the View - Properties - Precision dialog, they will be adjusted to topologically sensible positions.

This operator includes several functions that are executed simultaneously:

- Snap lines to nearest points - Used to prepare drawings for use as networks.
- Resolve overshoots and undershoots - Truncate overshoots and extend undershoots.
- Adjust area boundaries - Clip overlaps and extend areas to fill in gaps.
- Split lines at intersections.
- Remove redundant coordinates in object metrics that are unshared with other objects.
- Move defining coordinates to the nearest positions consistent with the Location Precision parameter and remove any redundancies.

This latter function leads to the use of Normalize Topology for "generalizing" drawings to a lower resolution by altering the Location Precision parameter in the View - Properties - Precision dialog of a drawing and then running Normalize topology.

Normalize Topology can handle any mix of points, lines and areas, including branched areas. Lines and points are pulled to neighboring areas and then lines and points are adjusted between themselves. Normalize Topology considers neighboring objects when making adjustments. It will automatically make fine adjustments so that small gaps do not appear between neighboring areas, for example.

See the Normalize Metric topic for illustrations of redundant metrics.

Use with Multiple Map Layers

To run Normalize Topology across map layers, run it on [All Objects] or on [Selection] if the selection includes objects in multiple map layers.

For better results (both speed and accuracy) make sure that all map layers and the map are in the same projection and all layers use the same location precision.

Optimization

Once Normalize Metric or Normalize Topology is run on an object, the object will be flagged within internal Manifold storage as clean. During future Normalize runs the process will skip examination and normalization of objects that are marked clean. If an object is subsequently modified (through editing by a user or by a script) the "clean" flag will be cleared so that the object will be examined by future passes of Normalize Topology. If the Location Precision property of a drawing is modified via the View - Properties - Precision dialog to a higher value (that is, a less demanding precision) then the "clean" flag for all objects will be cleared.

Because of the caching behavior of Normalize Topology, the amount of time required for the function will vary based on how many objects are been marked "clean." If the Location Precision has just been changed in a large drawing the function will take longer to run. If Normalize Topology has just been run (either explicitly or automatically as part of another transform), the function will run very rapidly since all objects will be marked "clean."

Comparison to Simplify

The Simplify command is similar to the Normalize Topology operator. However, the Simplify command is aimed at simplification on a per-object basis while the Normalize Topology operator considers relationships between objects when simplifying them. We can see the difference by running both commands on a drawing of Mexico.
We begin with a drawing of Mexico that's been projected into Orthographic projection.

Running Simplify with a Distance setting of 25000 meters results in the simplification of areas. However, each area is simplified without considering any relationships with adjacent areas. This provides an optimal simplification when each area is considered by itself but also results in overlaps and gaps between areas.

If instead of using Simplify we used Normalize Topology, we would first use View - Properties - Precision to set the Location Precision to 25000 meters and then we would run Normalize Topology.

Normalize Topology takes a lot longer to run than Simplify, but the result in the end is free of overlaps and gaps. Each area has been adjusted both on the basis of capturing its own shape as well as to match its neighbors. However, it could be said that the resulting shape for each individual object (without regard to neighboring areas) is not as "optimal" a simplification as is done by Simplify.

When simplifying a single object it is probably best to use Simplify since the result is obtained much faster. When simplifying many objects for which adjacency must be maintained it is best to use Normalize Topology.

Differences between Normalize Topology and Normalize Metric

Both of these transform operators are used to produce a clean metric; however, the metric produced by Normalize Metric may be slightly different than that produced by Normalize Topology. The main difference between them is that Normalize Metric is run on a per-object basis while Normalize Topology is run on an object set as a whole. When processing metrics Normalize Topology considers neighboring objects and Normalize Metric does not.
Examples:

- **Normalize Topology** will detect overlapping areas and will assign the region of overlap to one of the overlapping areas while clipping it from the others. In this same circumstance **Normalize Metric** will leave the overlaps.

- **Normalize Topology** will snap points to the nearest object (such as the end of a line) if the distance between the point and the object is less than the location precision. **Normalize Metric** will leave points unmoved.

- **Normalize Topology** will detect an intersection between two lines and will split the lines at the intersection. **Normalize Metric** will leave the lines unchanged.

Both **Normalize Topology** and **Normalize Metric** will remove redundant coordinates, but **Normalize Topology** may leave some redundant coordinates that would be removed by **Normalize Metric**. **Normalize Topology** will leave any redundant coordinates that are also coordinates in any other object. This is done to guarantee that a common border between any two adjacent objects is exactly the same in both objects. Note that **Normalize Metric** cannot do the same since it operates only on single objects.

How should one choose between **Normalize Topology** and **Normalize Metric**? In most cases, we would like any adjustments to object metrics to proceed with the entire object set in mind so in most cases one would use **Normalize Topology** in preference to **Normalize Metric**.

One of the rare cases where we might prefer **Normalize Metric** over **Normalize Topology** is when we must deal with a seriously inaccurate drawing that is known to contain many redundancies in object metrics, which we would like to quickly fix so that all subsequent operations are faster. In this case we could first run **Normalize Metric** at, say, one-fourth of the usual Location precision and then run **Normalize Topology** at the normal value of Location precision.

When dealing with imported drawings that may contain specific features of the metric that are to be preserved, users will often choose to begin operations by running **Normalize Topology**. This will clean object metrics while preserving redundant coordinates that are co-located with other objects, and it will mark all objects as having clean metrics. This avoids a situation where Manifold is compelled to run **Normalize Metric** even though we do not want it run.

If some other transform operator (such as **Clip Intersect**) is run on a drawing that has not been yet normalized with either **Normalize** function Manifold has no choice but to run **Normalize Metric** to guarantee a clean metric. By running **Normalize Topology** first we pre-empt the running of **Normalize Metric** by other transforms in self-defense. The main reason for running **Normalize Topology** first is to prevent the deletion of redundant coordinates that are co-located with other objects. A secondary (rare) reason is to prevent redundant coordinates from being deleted when we are working at a location precision that may cause coordinates to be judged "redundant" that are really needed to prevent gaps from emerging between formerly adjacent objects.

Why do the other transform operators run **Normalize Metric** instead of **Normalize Topology** if objects need cleaning before processing? Transform operators are frequently run on subsets of objects. If **Normalize Topology** were used automatically the usual workflow of running transforms on subsets of objects would quickly lead to a clustered drawing, different parts of which have been normalized with respect to different sets of objects. That would defeat the global normalization that is usually a key objective of **Normalize Topology**.

To summarize, when in doubt run **Normalize Topology** once before running other transforms. This can be very time consuming for large drawings, but it is a prudent and necessary step experts will take if they wish to preserve specific metrics.

**Tech Tip**

To prevent **Normalize Topology** from moving coordinates about (snapping points to the ends of lines, etc.), one simply sets the location precision to a very small value. The "small value" should be some value considerably less than any possible distance between distinct coordinates that occur in objects. For example, in a drawing originating from a USGS DLG where the normal accuracy is ten meters using a location precision of .001 meter will assure that no coordinates are moved by **Normalize Topology**. The procedure is:

- Launch the View - Properties dialog for the drawing and set Location Precision to a very small value.
- Launch Tools - Options dialog, go to Miscellaneous page and uncheck the Split branched objects after transforms checkbox.
See Also

Simplify
Transform - Normalize Metric
Transform - Relative Neighborhood Network

Given a set of points, builds a relative neighborhood network on that point set by adding lines.

A relative neighborhood network is a subnetwork of a Gabriel network (see the Gabriel Network topic). The links of a Gabriel network are drawn based on no points being within the circle defined by any pairwise point comparison, where the diameter of the circle is given by the distance between the points in the pair. In contrast, the relative neighborhood network draws a link if no other points appear in the area of intersection of two circles drawn centered on the points of the pair where the radius of the circles is the distance between the points of the pair.

In the figure above, the inner red circle shows the exclusion circle applied when drawing a Gabriel network. The area indicated in gray color shows the exclusion area applied when drawing a relative neighborhood network. Note that since the relative neighborhood network excludes links to nodes in a larger prohibited area, the relative neighborhood network will have fewer links than the Gabriel network.

The lines created by this operator will be selected after they are created. It's a good idea to move them to a new drawing to keep the map well organized.

Relative neighborhood networks are very useful because they not only consider the distance between two points considered as a pair, they also take into account the distance between that pair of points and all of the rest of the points. The two points are "relative neighbors" only if they are as least as close to each other as they are to all of the other points. Relative neighborhood networks therefore include a built-in reckoning of the overall arrangement of the point set in addition to local considerations. For this reason, they tend to reveal better cluster relationships than do simple local measurements such as nearest neighbors.

Example

Using the set of points above, we can create the relative neighborhood network seen below.
Note that a superset of a relative neighborhood network is created by the Gabriel Network transform operator as seen below:

See Also

**Spanning Tree** - The transform operator that creates minimum spanning trees.

**Clusters** - The transform operator that uses relative neighborhood networks to find clusters.

Notes

In graph theory, networks are called graphs. When searching Internet for information on these topics try searching for words like "relative neighborhood", "graph", "spanning tree", "cluster" and similar. These ideas are applied in an astonishing range of disciplines, from fungal spore distribution patterns to the characterization of finds in archaeological sites.

The points for the examples above were created by making centroids for provinces in Mexico using the Centroids (Weight) transform operator. The points and results of the transform operators are shown in a map with a drawing of Mexico as a backdrop. We used the centroids as a source set of points because they are dispersed in a geographically interesting way. There is no meaning to the map of Mexico; however, if we were to place one
antenna in the geographic center of each province and then we wished to link the antennas in a network we would likely create and study maps such as these.
Transform - Remove Duplicates

Remove duplicate objects. The **remove duplicates** operator deletes areas, lines or points that appear exactly at the same location. It is best used when field contents (if any) are not an issue, since this transform simply removes the most recently-created objects.
Transform - Reverse Lines
Reverse the order of coordinates that define a line, thus reversing the "direction" of the line. For multi-branched lines, the Reverse Lines transform will also reverse the order of branches so the first branch becomes the last and vice versa. Used with asymmetric line formatting styles so that the style is oriented as desired.

Directed Line Styles

Some line styles are asymmetric in that their appearance depends on the direction of the line. All lines in Manifold have a "direction" that is implied by the order in which they are drawn from the first coordinate that defines the line to the last.

A triangle line style is one such style. When looking along the direction of the line from the beginning to the end the triangles point to the left in this style. This can be seen by drawing a line.

We can use this style with the Insert Line tool to draw a new line by clicking at the left and then clicking at the right in the direction shown by the red arrow.

The result is a line drawn in triangles with the triangles pointing to the left along the direction of the line.

Suppose we draw a second line in the opposite direction as indicated by the red arrow.
In this case, the triangles will point in the opposite direction.

We can see that the orientation of the lines depends upon their direction, as shown by the red arrows. To switch the direction of a line (and thus the orientation of any asymmetric line style) select the line and use the Reverse Lines operator.

**Tech Tip**

Many transforms automatically run a Normalize Topology transform before running to eliminate common errors that may affect correct operation. The Reverse Lines transform operator does not normalize target objects before running.
Transform - Rotate (Drawings)

Rotate objects about the approximate center of their minimum enclosing circle by the number of degrees given in the source / argument box. Negative degree values cause counter-clockwise rotation while positive values cause clockwise rotation. Fractional values are allowed. For example, a parameter value of -45.5 rotates objects counter-clockwise by 45.5 degrees and a parameter value of 90.5 rotates objects clockwise by 90 degrees.

It is strongly recommended that drawings be projected before using this operator to provide maximum accuracy in rotation.

Rotation Example

We will rotate Germany within a drawing of Europe. The drawing has been projected into Orthographic projection.

![Maps showing rotation example](image)

We first select the area representing Germany. Next, load the Transform toolbar with

[selection] Rotate -45.5

and press Apply. This is a command to rotate Germany by 45.5 degrees counter-clockwise.

This command is most frequently used in CAD-style editing of non-geographic data. It is shown here in a geographic setting to make clear its effect.

Tech Tip

Many transforms automatically run a Normalize Topology transform before running to eliminate common errors that may affect correct operation. The Rotate transform operator does not normalize target objects before running.
Transform - Scale / Scale Horizontally / Scale Vertically

Resize objects by the scale given in the source / argument box. A parameter value of 0.5 resizes objects to half their size. A parameter value of 2 resizes objects to twice their size. The Scale Horizontally and Scale Vertically operators resize objects by the scale given in the source / argument box only in their horizontal (East / West) or vertical (North / South) extent.

It is strongly recommended that drawings be projected before using these operators to provide maximum accuracy in scaling.

**Scale Example**

We will resize Germany within a drawing of Europe. The drawing has been projected into Orthographic projection.

We first select the area representing Germany. Next, load the Transform toolbar with

```
[selection] Scale 0.5
```

and press Apply. This is a command to resize Germany to half the former size.

**Scale Horizontally Example**

We will now resize Germany only in its horizontal extent.

We first select the area representing Germany. Next, load the Transform toolbar with

```
[selection] Scale Horizontally 0.5
```

and press Apply. This is a command to resize Germany to half the former size in the horizontal (East / West) direction. All vertical extents will remain the same.

**Scale Vertically Example**

We will now resize Germany only in its vertical extent.
We first select the area representing Germany. Next, load the Transform toolbar with

[selection] Scale Vertically 0.5

and press Apply. This is a command to resize Germany to half the former size in the vertical (East / West) direction. All horizontal extents will remain the same.

These commands are most frequently used in CAD-style editing of non-geographic data. They are shown here in a geographic setting to make clear their effect.

Tech Tip

Many transforms automatically run a Normalize Topology transform before running to eliminate common errors that may affect correct operation. The Scale transform operators do not normalize target objects before running.
Transform - Segments

The **Segments** operator is used to insert additional coordinates into objects to allow preservation of shape when the objects are re-projected into a different coordinate system. The segmentize operator takes each straight line segment that comprises lines and areas and inserts additional coordinates, so that the single straight line segment will consist of many more straight line segments. These additional coordinates will be redundant in the existing coordinate system, but they may be necessary to preserve shape when re-projecting the drawing into a different coordinate system.

The transform toolbar version of this command adds the number of given segments, whereas the Segmentize command in the Drawing menu adds segments so that each segment is no greater than a given **Distance** parameter.

See the Segmentization topic for illustrations and explanation of the segmentization process.

**Tech Tip**

Many transforms automatically run a Normalize Topology transform before running to eliminate common errors that may affect correct operation. The **Segments** transform operator does not normalize target objects before running. Also, the **Segments** operator will not split resulting objects.
**Transform - Select Adjacent to**

Select all objects in the target set that are adjacent to any object in the source / argument box set. Objects are adjacent if they have one or more boundary coordinates in common with no other overlaps.

Suppose we have a saved selection called *Durango* that selects the province of Durango in Mexico.

Running **Select Adjacent** selects all other provinces that share a boundary with Durango.

**Tech Tip**

If **Select Adjacent** fails to select some "adjacent" areas but Select Touching does select those areas, carefully examine the region of adjacency to see if there is any overlap between the objects.

**See Also**

Select Contained / Containing
Select Intersecting
Select Touching
Transform - Select Contained / Containing

Select Contained within will select all objects in the target set that are entirely contained within an object in the source / argument box set. Select Containing will select all objects in the target set that completely contain any object in the source / argument box set.

Suppose we have a saved selection called Durango that selects the province of Durango in Mexico.

Select Contained within

Running Select Contained within selects the two points and the line that are entirely within Durango.

Select Containing

Suppose we have another layer in the map containing circular areas. Running Select Containing selects the large circular area that entirely contains Durango.
See Also

Select Adjacent to
Select Intersecting
Select Touching
**Transform - Select Euclidean Point Coverage**

Selects a minimal set of points so that each point from the source set is within the given distance to at least one point of the result set. The distance is given in the drawing’s native measurement units.

This is a very useful operator when locating antennas for some types of wireless services.

**Example**

Suppose we have a network created from roads in a town. The drawing is in Orthographic projection, which by default uses meters as the projection unit.

A point has been placed at the end of each line using the Points operator. Duplicate points have been removed with the Remove Duplicates operator. We could now delete the roads, since we no longer need them, but we will keep them for display purposes.

We can run the **Select Euclidean Point Coverage** operator in the transform toolbar on all objects using a parameter of 500. Because the drawing uses meters in its projection coordinate system, we are asking Manifold to construct the Euclidean point coverage using a distance of 500 meters.
The result of the operator is that four points have been selected and shown in red selection color.

If we draw circles about each red point that are 500 meters in radius, we can see that Manifold has selected four points such that every other point in the network lies within 500 meters of one of the four points.

The point coverage selected is **minimal**, because no fewer than four points can be selected such that all other points are within 500 meters of a selected point. We cannot choose only three points, for example, within 500 meters of all the other points. Four points are minimal, because five points are not necessary.

The point coverage is said to be **Euclidean** because the distances between points that are measured to be 500 meters or less are computed using straight line, Euclidean distance. They are not distances as measured through the network.

Note that the point coverage selected by Manifold is not guaranteed unique. There may be a different set of four points that also covers the other points to a distance of 500 meters. The guarantee is simply that there is no set of three or fewer points that is also a coverage to 500 meters of this particular network, and that the selected set of four points is a coverage to 500 meters.

Note that the illustration suggests a practical usage for this operator. Suppose we have a road network within a region that we would like to saturate with wireless service, such as wireless communications between centralized polling stations and reporting nodes. If each wireless antenna can cover a 500 meter radius and we would like to locate antennas only at nodes (to take advantage of the ease of trenching roads or pulling cable through existing conduits) we can use the **Select Euclidean Point Coverage** operator to find possible coverages with the minimal number of antennas required.

**Units of Measurement**

Units used for the distance value in the source / argument box are taken from the projection of the drawing. If the drawing is in **Latitude / Longitude** projection the units will be degrees. If the drawing is projected, the units will be in meters or feet. It is therefore strongly recommended that the drawing be projected first so that meters or other linear measure can be used instead of attempting to use degrees as a measurement unit.

To see which units are defined for the drawing's projection, click on the Tracker tool and make some measurements. Whatever units are used to show the result of the tracker tool are the units currently in force.

If the drawing uses degrees and meter-based creation of buffer zones are desired, first re-project the drawing into any convenient meter-based projection (such as **Orthographic**) so that meters will be used as the units of measure.
When this transform operator is used with a drawing layer within a map the units used by the drawing will be used. If the map component’s projection is degree based while the native projection for the drawing is meter based (as, for example, with the Orthographic projection) or vice versa, then the transform operator will use whatever units are used by the drawing’s projection.

See Also

Network User’s Introduction
About Networks
Drawing Networks
Not All Maps are Networks
Transform - Select Intersecting
Select all objects in the target set that intersect any object in the source / argument box set. Objects intersect if objects from both sets have some parts in common (excluding intersections consisting of common boundary coordinates only) and also have some parts not in common. Excluded, therefore, are cases where objects are entirely contained within other objects, or where some objects entirely contain others.

Suppose we have a saved selection called Durango that selects the province of Durango in Mexico.

Running Select Intersecting selects three lines that are partially inside Durango. Note that it does not select the two points and line that are entirely within Durango.

See Also
Select Adjacent to
Select Contained / Containing
Select Touching
Transform - Select Shortest Path

Given a selection that includes a system of lines that can be treated as a network plus two points this transform operator selects those lines that form the shortest path through the network between the two points.

Example

We begin with a map with three drawing layers:

- A background map of Mexico. This has been set to low opacity so it is faint and unobtrusive.
- A drawing called Nodes that contains points. These were created for this example by running the Centroids transform operator on the Mexican province areas.
- A drawing called Links that contains lines. These were created for this example by running the Relative Neighborhood Network transform operator on the points in the Nodes layer.

In the Nodes layer we select two points. If we like, we can save this selection in the Selections pane so we can later select exactly these two points again.
In the Links layer we select all of the lines. This may be done rapidly by enabling Edit - Select Objects - Lines (or simply pushing in the Select Lines button) and then choosing Edit - Select by Type.

Run the Select Shortest Path transform using the [Selection] as the scope. Press Apply to select those lines that comprise the shortest path between the two points through the network.

The result is that the lines on the shortest path are selected, and highlighted in red selection color. (In the illustration above we have also used the saved selection to select the two points as well.) We will often save this selection, or copy the lines involved and paste them into a new drawing.

**Troubleshooting**

When running this transform operator and getting unexpected results, check the following:

- Are you using an object set that contains only the two points between which you want a shortest path? Running this operator on a set of objects that contains more than two points will select two random points and select a shortest path between them.
- Do the lines in use form a network? That is, are they contiguous or are there breaks in the lines? Use Normalize Topology to make sure the various ends of lines are coincident with the beginnings of the next lines.
- Are the points in use coincident with the ends of the lines? If not use Normalize Topology to make sure they are coincident.

**See Also**

**Not All Maps are Networks**
Transform - Select Service Centers

Manifold includes four transform operators to find optimal locations for service centers. These include:

- Select Critical Service Centers
- Select N Critical Service Centers
- Select Service Centers
- Select N Service Centers

The operators solve two types of network tasks: to find regular service centers and to find critical, or emergency service centers. Both types of task occur in two forms: to find just one location for a service center or to find an \( N \) number of that type of service center.

**Regular Service Centers**

Given a set of nodes, called clients, finds the "central" node that minimizes the path length to travel from the center to each of the clients and to then return to the center. This is the classic problem for locating a regular or daily service center, where every day one must make the rounds of each of the clients. A real life example would be locating an industrial bakery that every morning must send a truck to deliver bread to the stores and restaurants that resell its baked goods. Every day the truck visits the same clients. The job is to place the bakery so that the overall distance travelled is minimized.

**Critical Service Centers**

Given a set of nodes, called clients, finds the "central" node that minimizes the maximum path length to any of the clients without regard to return travel. This is the classic problem for locating an emergency service center, where at any time one may be called upon to travel to a given client. A real life example would be locating a fire station that is equipped with regional facilities for fighting certain types of hazardous chemical fires. The possible clients are those factories that use such hazardous chemicals, but it is not known in advance at which factory an accident requiring the specialized equipment will occur. Thus the task is to place the fire station to minimize the distance to any one of the factories at which an accident might occur.

**Variations**

Clearly, both of the above examples are somewhat artificial. However, the general idea of minimizing distance for regular service or minimizing distance for emergency service can be applied to many network problems besides the siting of bakeries or fire stations. The Manifold transforms are deliberately kept fast and simple so they are easy to use. The main option is the use of the source / argument box in the Transform toolbar to specify the number of different service centers that are to be found. The \( N \) versions of the operators will find a dispersed pattern of the given number of service centers, placing each to optimize the service overall. The non-\( N \) version of the operators assumes we really intend to place only one service center, but it provides several backup options that are less optimal near the central location found.

**Requirements**

All four of the service center solvers require a connected network to function. In a connected network each point is exactly placed at the end of a line and all lines that are incident to other lines have their end coordinates exactly coincident. There are no gaps, points that are not located exactly on the lines, or lines that are intended to be incident but where the ends don't exactly coincide.

**Examples**

We begin with drawing of a network that has nodes at the ends of all links.
To find the four best locations for regular service centers, we run the **Select N Service Centers** operator with a parameter of 4 in the transform toolbar.

The operator selects the four optimal nodes. We could then copy these nodes and paste them into a new drawing if we wanted to record our findings.

The selection made by the **Select Service Centers** operator run with a parameter of 4 is seen above. The assumption is we are looking for only one regular service center but we want to have a choice of the four best candidates.

Running the **Select N Critical Service Centers** operator with a parameter of 4 creates the selection above. Note that although the nodes selected are similar to those selected for the regular centers, there is one node that is different from the similar node selected by the regular centers algorithm (blue arrow).
The Select Critical Service Centers operator with a parameter of 4 provides very different results than the regular service centers operator. These are the best four choices if an unpredicted dispatch must be made to any other node.

If the results for four centers seem counter-intuitive, consider the result for a parameter of 2. The lower of the two selected nodes is the best (determined by running the transform with a parameter of 1), while the upper one is second best. This shows the power of graph theory to reveal solutions that one might not pick out by eye.
Transform - Select Touching
Select all objects in the target set that touch any object in the source / argument box set. Objects touch if they either intersect or are adjacent to one another.

Suppose we have a saved selection called Durango that selects the province of Durango in Mexico.

Running Select Touching selects the two points and line that are entirely within Durango as well as three additional lines that are partially inside Durango. It also selects the areas that are adjacent to Durango.

See Also
Select Adjacent to
Select Contained / Containing
Select Intersecting
Transform - Shape Hull

Tile objects in the drawing with minimum enclosing polygonal areas built up out of square tiles using the grid specified in the parameter. Use a value greater than 1 in the parameter.

Given the two lines above, applying the Shape Hull transform operator with a parameter of 50 creates the area shown below. Decreasing the parameter results in a more stair-stepped, coarser enclosing area while increasing the parameter results in a finer enclosing area.
Transform - Spanning Tree
Given a set of points, builds a minimum spanning tree network on that point set by adding lines.

A spanning tree is a network that connects all of the points in the point sets without forming any loops. A minimum spanning tree connects points so that if the lengths of all the links are added up there is no way to connect all of the points using links with a lower total length. If we wish to connect all the points using fiber optic cable the minimum spanning tree will use the least total amount of cable.

Although points sets will often have one and only one spanning tree arrangement that results in the lowest distance, regularly arranged point sets (such as those in orthogonal grids) can have many different spanning tree networks drawn that will all have the same, lowest, total length of links. The minimum spanning tree transform simply guarantees that no tree can be drawn that uses less material than the spanning tree it finds.

Given the point set above, the minimum spanning tree operator would create the network shown below:

Minimum spanning trees are very important in a very wide range of disciplines. Because they show how to connect points with the least connecting material they are obviously important in any analysis that requires short distances between connected items and minimization of overall connection distances.
Given a network, this operator selects those lines in the network that make up a minimum spanning tree.

We can create the network above by inserting points into a drawing and then running the Relative Neighborhood operator.

If we then run the Select Spanning Tree operator it will select those lines that participate in a minimum spanning tree.

See Also

Clusters - The transform operator that in the Clusters (Zahn) form uses minimum spanning trees to find clusters.

Relative Neighborhood Network - The transform operator that creates relative neighborhood networks, which are subsets of Gabriel networks. This topic includes a note on characteristics of relative neighborhood networks.

Notes

In graph theory, networks are called graphs. When searching Internet for information on these topics try searching for words like "relative neighborhood", "graph", "spanning tree", "cluster" and similar. These ideas are applied in an astonishing range of disciplines, from fungal spore distribution patterns to the characterization of finds in archeological sites.

The points for the examples above were created by making centroids for provinces in Mexico using the Centroids (Weight) transform operator. The points and results of the transform operators are shown in a map with a drawing of Mexico as a backdrop. We used the centroids as a source set of points because they are dispersed in a geographically interesting way. There is no meaning to the map of Mexico; however, if we were to place one
antenna in the geographic center of each province and then we wished to link the antennas in a network we would likely create and study maps such as these.
Transform - Spline
Smoothes lines and areas by inserting additional midpoints and using a spline algorithm to replace sharp corners with curves. The value in the source / argument box determines the number of midpoints inserted into each segment.

Example

Consider a drawing containing a line and an area.

After running the Spline operator on [All Objects] with a source / argument box value of 5 both the line and the area have been smoothed.

Smoothing will preserve common boundaries between adjacent areas.
For example, suppose we have two adjacent areas like the above.

Running the Spline transform (again, with a parameter value of 5) creates the above smoothed boundary between the two areas.

**Notes**

*Spline* is a great way to provide a more natural, rounded appearance to vector drawings in GIS. It is similar to how Bezier curves are used to provide rounded curves.
Transform - Split With

Split target areas and lines using a given set of lines. The transform takes the given set of lines and cuts into separate objects any line and area objects that are intersected by the lines.

Example

Suppose we have a drawing with an area and a line.

We select the line and in the Selections pane save this selection as a saved selection called line.

We next select the area and in the Selections pane save this selection as a saved selection called area.

We can now set up the transform toolbar with the Split with operator as seen above and press Apply

The result is that the area object is split into two areas, neatly cut by the line. We can select the rightmost area.
If we press **Delete** the area disappears, proof that what was once one area was split into two areas.

**Comments**

The **Split with** operator uses any set of lines to split any combination of areas or lines. The example above shows just one line being used to split a single area. We can use the transform in far more significant ways. For example, we could use lines showing streets to split a background area and thus create areas within the various streets that could be used as city blocks, or we could use the boundary of Switzerland to "cookie cutter" VMAP 1 lines and areas that cross the border.
Transform - Triangulation

A Delaunay triangulation of a point set treats the points as nodes in a network and draws links between them that divides the region between the points into triangular tiles.

If we take a set of points as shown above and apply the Triangulation Lines transform operator we can create a triangulation consisting of lines.

The result is a set of lines that show the boundaries of triangular tiles that completely cover the region between the points. Had we wished to create tiles in the form of area objects, we could have used the Triangulation Areas transform operator. Using the Triangulation operator would simultaneously create both Triangulation Lines as well as Triangulation Areas.

There are many different algorithms that may be used to decide how a point set should be triangulated. Manifold's transform toolbar Triangulation operators use Delaunay triangulation (also spelled the Delone triangulation). The Delaunay triangulation is closely related to the Voronoi tiling of a region, as can be seen from the following illustration that shows both a Voronoi tiling as well as a triangulation.

The blue lines show the borders of Voronoi tiles. The green lines show the Delaunay triangulation. To make the triangulation we draw a line between every two points that share a border line in the Voronoi tiling.

Triangulations can be used for many purposes. They are a natural way of creating a network by connecting points that allows "travel" between points. Triangulations have great use in interpolation as well.
For example, if we begin with a set of points shown above we can create a triangulation as seen below:

Transfer Rules

The triangulation operators except will transfer column data from source to target (created) objects using whatever transfer rules are in force for the data attribute columns.

See Also

Decompose to Triangles
Transform - Constrained Triangulation

Other types of networks easily created with transform operators:

Gabriel Network
Relative Neighborhood Network
Spanning Tree

Historical Note - A Tale of Two "Delaunays":
Through an accident of translation of a Russian name into Latin characters, the wrong mathematician is often credited with the invention of "Delaunay" triangulation.

Charles-Eugene Delaunay was the French mathematician and astronomer who often is given credit for the triangulation method bearing this name. However, he is the wrong "Delaunay".

Charles-Eugene Delaunay (1816 - 1872)

Educated at the Ecole des Mines in engineering and at the Sorbonne in astronomy, the French mathematician and astronomer Charles-Eugene Delaunay is best known for his contributions to the theory of lunar motion. He is honored with a lunar crater named for him as well as several street features in Paris. These include the Square Delaunay, the Rue Delaunay and (perhaps especially amusing to GIS beginners) the Impasse Delaunay. Delaunay drowned in 1872 in a boating accident in the English Channel near Cherbourg.

Boris Nikolaevich Delone (pronounced "Delaunay" and often spelled that way in English as well) is the Russian mathematician who invented the triangulation method now universally used throughout computational geometry.

BORIS NIKOLAEVICH DELONE

Boris Nikolaevich Delone (1890 - 1980)

Boris Delone was born in St. Petersburg on March 15, 1890 and lived a long life as a mathematician and mountain climber. He persisted in the study of algebra even after the utilitarian transformation of society in the wake of the October revolution discouraged the study of abstract mathematics. After graduating from Kiev University in 1913 Delone taught at the Kiev Polytechnic Institute. He moved to St. Petersburg in 1922 to join the faculty at Leningrad University. In 1932 he worked in the Mathematics institute of the Academy of Sciences. In 1935 he became a professor of Mathematics at the University of Moscow (MGU) from 1935 to 1942. His work in triangulation arises from his work in mathematical crystallography. He also worked in computational geometry, the theory of numbers, and the history of mathematics as well as continuing his life-long researches in algebra. Delone became an Academician in 1929.

Delone's fame as a mountain climber within that sport was equal to his fame as a mathematician in scientific circles. He climbed numerous peaks of the highest difficulty in the wilds of the Caucasus, Central Asia and the Altai. He wrote of his life in mountaineering: "Mountain climbing in my life was not simply a sport or the source of a good mood. It is a worldview that asserts simple truths, glorifying the good things: bravery and comradeship, the desire to know and the desire to help, a devotion to purpose, a sense of and joy in daring, keenness and striking courage ".

Pronunciation

"Delaunay" is normally pronounced in the French style, with stress on the final syllable: "Deh - lah - NAY". The name is pronounced with "short" vowels in the first two syllables and a long "a" in the last syllable.
The Cyrillic for "Delone" is pronounced the same way as "Delaunay". It is an oddly French-sounding surname within the Russian world. Due to pronunciation conventions in English it strikes us as odd that the Latin spelling of "Delone" would be so pronounced. For this reason it is often spelled "Delaunay" in English texts, the convention adopted here.
**Transform - Union**

The **Union** operator in the Transform toolbar for drawings combines into a single area all areas in the target set. If the areas being combined do not touch, a multi-branched area object will be created.

In the illustration above, four Western US states areas have been combined into a single area object using **Union**.

**See Also**

**Dissolve** - A method of combining objects using the values of data attributes.
Transform - Voronoi Operators

These operators create a Voronoi diagram for the points in the active drawing.

Suppose we have a drawing of points. A Voronoi diagram divides the drawing into regions around each point that are shaped so that the borders of the regions are equidistant from the two nearest points.

By drawing lines to mark out Voronoi cells (or drawing areas in the shape of those cells), we divide up (or tile or tessellate) the drawing into regions. Every location within a Voronoi cell is closer to the point about which that cell is drawn than it is to any other point. Voronoi diagrams are very important for dividing drawings into regions associated with points.

- **Voronoi Diagram**: Create area, line and point objects for each Voronoi cell.
- **Voronoi Areas**: Create area objects for each Voronoi cell.
- **Voronoi Lines**: Create line objects at the border of each Voronoi cell.
- **Voronoi Points**: Create point objects at the intersections of the borders of the Voronoi cells. Rarely used.

The illustration seen above shows the effect of the Voronoi Lines operator.

The Voronoi Diagram operator, as seen above, creates area, line and point objects for the Voronoi diagram. Note that points appear at the intersection of the borders of the Voronoi cells. These points are created by the Voronoi Points operator.

The areas, lines or points created by this operator will be selected after they are created. It's a good idea to move them to a new drawing to keep the map well organized. Move them by using Edit - Cut and then Edit - Paste into a new drawing or Edit - Paste As a new drawing in the project pane.

**Example**

Suppose we have a few hundred environmental sampling stations scattered throughout a region. We also have a dozen data collection centers, numbered 1 through 12, within the region.
We would like to assign each sampling station to the nearest collection center. To do this we first use a drawing of the data collection centers to create a **Voronoi Area** surrounding each center.

![Voronoi Diagram](image)

We can then use the Spatial Overlay dialog to transfer the identification number of each data collection center to the Voronoi cell that encloses it. Next, we can use **Spatial Overlay** once more to transfer the identification number from each Voronoi cell to all of the sampling station points within each cell. The result is that each sampling site will have a field that contains the data collection center number that services it.

**Transfer Rules**

All of the Voronoi operators except **Voronoi Lines** will transfer column data from source to target (created) objects using whatever transfer rules are in force for the data attribute columns.

**Historical Note:**

Voronoi diagrams are also known in some cultures as Dirichlet or Thiessen tessellations. Although individual investigators have used this powerful concept informally at least as far back as Descartes in 1644 the key researchers formally developing this concept were Dirichlet and Voronoi.
Dirichlet used a special form of the Voronoi tessellation in his study of positive quadratic forms. Dirichlet was born in a part of the French Empire long disputed back and forth between France and Germany, studied in Paris and settled down to an overworked and productive career in Germany. Voronoi later published a generalization of Dirichlet's concept that would apply to higher dimensions and so introduced the concept in its modern form.

Georgi. F. Voronoi (1868 - 1908)

Voronoi was born in Russia on 28 April 1868 and graduated from the University of St. Petersburg in 1889, winning the Bunyakovsky prize for his Master's thesis and again a second time for his Doctor's thesis. He was a lecturer at Warsaw University and contributed to the theory of algebraic numbers and the geometry of numbers.

At times Voronoi wrongly is claimed to be a German mathematician (an error repeated in some web sites). Even more inaccurately, some people refer to Voronoi's work by crediting the concept to Thiessen, a German meteorologist. Both errors appear to arise from the dominance during the inter-war years of German researchers in crystallography and other subjects in which Voronoi diagrams are used.

Thiessen used the idea of Voronoi diagrams much later than either Dirichlet or Voronoi, beginning only in 1911 to apply them to the study of meteorology. Thiessen quite likely felt he had independently derived the concept (as have many workers in the years since Voronoi's publications).

As a meteorologist Thiessen probably would not have had full awareness of all that was done in mathematics by professional mathematicians such as Dirichlet or Voronoi. But is unfortunate that some writers who certainly knew of Voronoi's work would deny either Voronoi or Dirichlet credit while advancing Thiessen as the inventor of a concept that Voronoi developed both more fully and at an earlier date, not to mention Dirichlet's earlier work as well.

Whatever the reason for the original misattribution, in modern times if we are not to inadvertently repeat the error we should use the term "Voronoi" or "Dirichlet" tessellations for this concept. The term "Voronoi" is used by Manifold because it was Voronoi who presented the mathematics of this notion in the contemporary form used within Manifold.

Pronunciation

"Voronoi" is pronounced by English speakers as "Vo - ro - noi" with a short "o" sound, like the "o" in "or", for the first two syllables. The third syllable is pronounced like the "noi" in "noise". The stress is on the third syllable. Russian speakers will pronounce the name with such a short "o" in the first two syllables that it sounds like "uh" or even "ah".
The Business Tools package is an optional extension to Manifold System that provides additional commands for working with drawings. The extension adds functionality in several areas:

- **Districts (Advanced)** - Redistricting using a condensed, expert level dialog.
- **Districts (Visual)** - Redistricting via an easy-to-use visual, interactive dialog.
- **Drive-Time Zones** - Computation of drive-time zones via an easy-to-use, interactive dialog.
- **Optimal Route** - Finding an optimal route to designated locations using a condensed, expert level dialog.
- **Optimal Route (Visual)** - Finding an optimal route to designated locations via an easy-to-use, interactive dialog.
- **Send Email** - Automatically send email to map objects. Used to send spatially-targeted email.
- **Topology Factory** - An interactive dialog to view and repair common topological errors in drawings.
- **Expanded programmatic access to routing and other functions.**

If you do not have the optional Business Tools package enabled you will not have the above capabilities enabled within Manifold System. This documentation describes these commands but they will not be enabled if you do not have the Business Tools package enabled.

### Installing Business Tools

The Business Tools package is built into your Manifold System installation and becomes enabled for use when activated. Activate it by providing a Business Tools serial number and Activation key that turn on Business Tools functionality. The process of activating the Business Tools extension is very similar to that used to activate Manifold System. See the Activation Keys and Serial Numbers topic and the Installing and Activating a Manifold Extension topic for details.

Some Manifold product options, such as Universal Edition or Ultimate Edition, use a single serial number and Activation key to enable permanent installation of both Manifold System as well as extensions. If you have activated Universal Edition or Ultimate Edition you do not need to individually turn on Business Tools or Geocoding Tools or Surface Tools. Activating Universal Edition or Ultimate Edition will also automatically turn on all three extensions.

### Turning on Business Tools

If you have licensed Universal Edition, you do not need to turn on Business Tools. The Business Tools extension is automatically enabled with a Universal Edition serial number.

If you have not licensed Universal Edition and you wish to add Business Tools functionality to your installation, begin by acquiring a Business Tools license from manifold.net, which will provide you with a Business Tools serial number.

2. Launch Manifold System. Close any projects that may be open.
3. Choose Help - Activate Extension
4. Enter the Business Tools serial number. Enter it exactly as it was issued to you by manifold.net. Do not change upper case to lower case. Do not replace hyphens with space characters or make any other changes to the serial number. If desired, you may also enter an Activation key for the serial number as well. Press Accept.
5. Manifold will then exit. When re-launched, the Business Tools package will be enabled.
6. Using the serial number alone the Business Tools package may be run for 30 days from the date the serial number was issued. After that, an Activation Key must be fetched and provided to the Help - Activate Extension dialog together with the serial number. Once an Activation Key and a serial number have been provided to the Help - Activate Extension dialog the extension will be permanently installed. Don't forget to login as Administrator when permanently activating the Business Tools extension.

### Status of Extensions
The Help - About dialog shows all extensions that have been installed. If a serial number only has been used to install an extension, the number of days left before an Activation key is also required will be shown.

**See Also**

- Districts (Advanced)
- Districts (Visual)
- Drive-Time Zones
- Optimal Route
- Optimal Route (Visual)
- Send Email
Districts (Advanced)

The Drawing - Districts (Advanced) menu selection is enabled when the optional Business Tools extension has been installed and the focus is on a drawing window or on a drawing layer in a map window that contains areas. If you have not activated the Business Tools extension with a valid Business Tools serial number you will not be able to use the Districts (Advanced) command.

The Districts (Advanced) command takes a set of areas and automatically assigns each area into as many districts as desired. For example, if four districts are desired each area in the set will be assigned to one of the districts. Areas will be assigned so districts will be made up of areas that are grouped together into the district as reasonably as possible. The districts commands in Manifold are used to create districts that represent sales territories, voting districts or other groupings of areas, which might be states, postal code areas, voting precincts or other regions.

The Districts (Advanced) command will also assign areas based on the content of a numeric data field so districts can be balanced by the sum total of that field. For example, areas can be assigned into districts based on the content of a Population field so that the resultant districts all have approximately the same population. Options allow balancing not only so that each district has the same total value of the balancing field but that some districts can have proportionately more or less. For example, the command may be used to create four districts, one of which has twice the population of all the rest.

Using this command, Manifold can either create districts by assigning areas to them without any preconditions, or we can specify which areas must be in which district.

The Districts (Advanced) command is a text dialog-driven command that works with values in the drawing’s table. For a more visual, interactive districting function, see the Districts (Visual) command.

Controls

<table>
<thead>
<tr>
<th>Control</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scope</td>
<td>A set of areas to be assigned into districts. Normally [All Objects] to assign all areas in the drawing but can be the selection or any saved selection.</td>
</tr>
<tr>
<td>Balance</td>
<td>A numeric field name to use for balancing districts. Choose none to balance districts by no field but just by the number of areas in each.</td>
</tr>
<tr>
<td>Save to</td>
<td>The name of a numeric column to use for district identification numbers.</td>
</tr>
<tr>
<td>Rebalance existing districts</td>
<td>Enabled if an existing column is selected in the Save to box. Check to rebalance districts given by the identification numbers in that column.</td>
</tr>
<tr>
<td>Modify formatting</td>
<td>Check to modify the formatting of areas so that areas in the same district appear in the same colors.</td>
</tr>
<tr>
<td>Create contiguous districts</td>
<td>Check to create contiguous districts.</td>
</tr>
<tr>
<td>Districts</td>
<td>A sequence of numbers separated by commas giving the number of districts to be created and the relative weights given to each district.</td>
</tr>
</tbody>
</table>

To assign areas into districts:

1. Open a drawing that contains areas.
2. Choose [All Objects] to assign all areas in the drawing to a district.
3. If districts are to be created based on some data attribute field, choose the field to use in the Balance box. If not, choose none.
4. Choose a column in the Save to box into which each district identification number will be written. The default choice of [New Column] will create a field called District and will write the district number for each area into that column.
5. If formatting is to be modified to show each district in a different color, choose Modify formatting (the default).
6. In the **Districts** box enter a number followed by a comma for each district to be created. For example, the default of "1, 1" tells Manifold to create two districts. To create four districts we would enter "1, 1, 1, 1" in this box. Press **OK**.

The command will add a column called **District** to the drawing’s table and place the district number in that column for each record. District numbers are integers beginning with 1 so that after the **Districts (Advanced)** command operates all areas in the first district will have the number 1 in the **District** column, all areas in the second district will have the number 2 in the **District** column and so on.

If the table already has a **District** column the **Districts (Advanced)** command will use the values in the column to control its activity. All area records with a 0 in the **District** column will be assigned to districts. All area records with a non-zero value in the **District** column will retain that value as their district assignment. For example, if some areas have a value of 1 and others have a value of 2 with all the remainder having 0 then when we run the **Districts (Advanced)** command to create two districts those areas having a value of 1 will be assigned to district 1, those with a value of 2 will be assigned to district 2 and all other areas will be assigned to either district 1 or 2. Placing pre-existing values in a **District** column is a handy way of forcing some areas to be assigned to given districts.

The series of numbers separated by commas in the **Districts** box specify how many districts to be assigned and also specify the relative weight for each. For example, a 1, 1, 1 string tells Manifold to create three districts, each of which is approximately the same size considering the total value of the field specified in the **Balance** box. A 1, 1 string tells Manifold to assign two districts of equal size and a 1, 1, 1 string tells Manifold to create four districts of equal size.

The size of the number in the series specifies how large that district should be relative to the others. For example, a string of 1, 3.5, 1, 1 tells Manifold to assign four districts, the second of which is to have three and a half times the total value of the other districts. A string of 1, 2, 2, 2 tells Manifold to create four districts, the first of which is one half the size of the others. Note that the values are purely relative. We could express the same effect with 0.5, 1, 1, 1 or 2, 4, 4, 4.

**Example**

Consider a drawing showing a group of adjacent areas. The example drawing shows areas formed of triangles and other straight edged polygons, but it could just as easily be an area showing countries, postal code districts or some other real geographic drawing.

![Areas](image)

Our task is to assign each area in the drawing to one of four districts. Ideally, we would like the districts to be reasonably compact in shape, so that areas in the same district are adjacent to each other.
If we open up the drawing's table we can see that it has no columns in it except the default system ID column.

With the focus on the drawing, we launch the Districts (Advanced) dialog. We will balance using the intrinsic field Area (I), which will result in districts all having approximately the same area. We use the string 1, 1, 1, 1 in the Districts box to create four districts of the same size. We leave the Modify formatting box checked (the default) so that right away we can see how the areas are assigned into different districts. Press OK.

Right away, the drawing is reformatted to show the new districts that have been assigned. Manifold has attempted to assign areas to districts so that the total area of each district is approximately the same and that areas assigned to each district are adjacent to each other.
If we look at the drawing's table we can see that a new District column has been added to the table. Each value in the District column specifies to what district that area object has been assigned.

For example, if we select one of the areas in the drawing…
…we can see it has a value of 3, meaning it has been assigned to district number 3.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>IC</td>
<td>555</td>
</tr>
<tr>
<td>District</td>
<td>3</td>
</tr>
</tbody>
</table>

If we would like to change the assignment of the area, we can double-click on the area in the drawing and in the Object Fields dialog change the value of the District field to 1.

The formatting of the object will change to the color for district 1...

<table>
<thead>
<tr>
<th>ID</th>
<th>District</th>
</tr>
</thead>
<tbody>
<tr>
<td>552</td>
<td>1</td>
</tr>
<tr>
<td>553</td>
<td>1</td>
</tr>
<tr>
<td>554</td>
<td>1</td>
</tr>
<tr>
<td>555</td>
<td>1</td>
</tr>
<tr>
<td>556</td>
<td>1</td>
</tr>
<tr>
<td>557</td>
<td>3</td>
</tr>
<tr>
<td>558</td>
<td>1</td>
</tr>
<tr>
<td>559</td>
<td>3</td>
</tr>
<tr>
<td>560</td>
<td>3</td>
</tr>
<tr>
<td>561</td>
<td>1</td>
</tr>
<tr>
<td>562</td>
<td>1</td>
</tr>
<tr>
<td>563</td>
<td>1</td>
</tr>
<tr>
<td>564</td>
<td>3</td>
</tr>
<tr>
<td>565</td>
<td>3</td>
</tr>
<tr>
<td>566</td>
<td>3</td>
</tr>
</tbody>
</table>

…and the value in the table, of course, will also now be 1.
Although the Districts (Visual) dialog is much easier to use for interactive changes in the assignment of areas to districts, we can show some more intricate work with the Districts (Advanced) dialog to illustrate how the assignment of areas to districts is nothing more than the assignment of a District number to each record.

For this example we will take four areas that are each assigned to a different district, and then reorganize all the other areas into districts about them. We being by selecting four areas, one from each district, which we want to force to be in each of the four districts.

In the table we can see that some of the records have been selected.
If we choose **Edit - Select Inverse** (or simply press **CTRL-I**) we can invert the selection so that all records previously unselected are now selected.

The table now shows that all records except the four areas we chose are now selected. We did this so we can change the **District** value in each of these records to 0. To do so, double-click into the **District** cell of any of the selected records and change the **District** value to 0.
All of the District values in the selected records will also change to zero. This is a very useful facility within Manifold that often comes in hand. We've deselected the records to provide a clearer illustration.

If we take a look at the drawing we see that there now remain only four areas assigned to areas with all of the other areas formatted in default gray.

We now launch the Districts (Advanced) dialog once more. This time we will use a Districts string of 1, 1, 2.5, 1, which tells Manifold to create four districts where district 3 is two and a half times larger than the other districts. In executing this command, Manifold will be constrained by the four areas we have already assigned to districts 1, 2, 3 and 4 because of the non-zero values they already have in their Districts column.
When we press OK we can see how Manifold copes with the command. It has assigned areas so that districts 1, 2 and 4 have about the same area and district 3 has approximately two and a half the area of the other districts.

Examining the table, we can see that those areas which previously had 0 in the Districts column now have numbers from 1 to 4 in the Districts column, representing their assignment to one of the four districts.

**Notes**

In the above examples we balanced districts using the Area (I) intrinsic field. This balances the total area of each district. If the drawing's table had other fields, such as the population of each area, we could have used one of those fields to balance the districts.

In "real life" situations we often want to use fields such as population or total sales to balance areas such as counties or zip code regions to form districts. In a law enforcement application we might want to assign territories to detectives such that the total number of crime incidents is balanced (the objective being so that the investigative workload on each detective's territory is about the same). In a public health application for medical services in rural regions, we might want to assemble districts to balance population, so that the medical staff assigned to each district has the same workload as other districts.

In such applications, quite often it is the case that some areas must be assigned to given districts because the district headquarters is already located there. We can use techniques like the example above to force some areas to be assigned to given districts and to then let Manifold optimize the allocation of other areas between districts.
Manifold will do its best to balance districts as directed; however, given the granularity of the balancing field and the constraint to have reasonably connected districts it may not be possible to have districts that are exactly balanced given the proportions desired.

See Also

Districts (Visual)
The **Drawing - Districts (Visual)** menu selection is enabled when the optional Business Tools extension has been installed and the focus is on a drawing window or on a drawing layer in a map window that contains areas. If you have not activated the Business Tools extension with a valid Business Tools serial number you will not be able to use the **Districts (Visual)** command.

The **Districts (Visual)** command provides an interactive dialog that allows specifying districts using a point and click visual user interface. In the main display pane the dialog shows a visual preview of the drawing in which districts are assigned. The dialog also lists the districts and relevant data for each district in a districts list pane.

To operate the dialog, add the number of districts desired into the district list pane (which opens by default with two districts), choose a field to use for balancing the districts and press **Assign**. Districts may then be modified by clicking as desired in the visual preview pane or in the districts list pane. Press **OK** to apply the districts to the drawing and **Cancel** to exit the dialog without making any changes.

To make manual adjustments, choose one of the districts in the districts list pane by clicking on that row to highlight the district. Clicking on an area in the visual preview pane will now assign it to that district, or if it is assigned will remove it from the district (clicking an area will toggle its membership in the district). The toolbar buttons just above the districts list pane affect the highlighted district in the visual districts pane. For example, to remove a district from the list, click on it to highlight it and then press the **Delete** button.

**Controls**

- **(Visual preview pane)** Displays the drawing either in default gray color or as colored by districts assigned with the **Assign** button.
  - **Zoom To Fit** - Zoom so that the entire drawing fits within the preview pane.
  - **Zoom In** - Magnify the view as if seen from a closer distance.
  - **Zoom Out** - Reduce the view as if seen from farther away.
  - **Zoom Box** - Zoom to the size of the cursor box drawn with the mouse.
  - **Center Point** - Pan the view so that the spot clicked is centered.
**Grabber** - Interactive pan: click and drag with the grabber hand. The scene will be panned so that the initial point is moved to the spot where the drag is released.

**(Districts list pane)** A list of districts to which areas may be assigned. The first number gives the relative weight of each district. Use the default weight of 1.00 so that each district will be created the same size (that is, the same total value of the Balance field). Click on a district to highlight it. Double-click into either the weight cell or into the color well to change the weight or color used for that district.

**New** - Add a district in the district list pane.

**Delete** - Delete the highlighted district from the districts list pane.

**Move to Top** - Move the highlighted district to the top of the district list pane.

**Move Up** - Move the highlighted district up one position in the district list pane.

**Move Down** - Move the highlighted district down one position in the district list pane.

**Move to Bottom** - Move the highlighted district to the bottom of the district list pane.

**Fill** - Assign all unassigned areas to the highlighted district.

**Clear** - Toggle to unassigned all areas in the highlighted district.

**Clear All** - Clear all district assignments from all areas.

**Show Counts** - Push in to show the number of areas in the district for each district in the districts list.

**Show Values** - Push in to show the sum total of the Balance value for all areas in the district for each district in the districts list.

**Show Value Percentages** - Push in to show the percentage of the total value for all areas being districted of each district in the districts list.

**Assign** Assign all unassigned areas to districts using the given settings. Areas that are already assigned to districts will not be re-assigned.

**Scope** Objects to be districted. [All Objects] or [Selection] or a saved selection.

**Balance** Choose a numeric field to use to balance districts. Manifold will add up the totals in this field when assigning areas so that each district has the same total value for this field. Choose none to balance districts by no field but just by the number of areas in each.

**Save to** The name of a numeric column to use for district identification numbers.

**Load Districts** Enabled if the drawing already has a numeric field called District or some other field that can be used to store district numbers. Pressing the Load Districts button will load district assignments into the dialog from the values in
the District field or from whatever field has been chosen in the Save to dialog. This is used when districts have already been assigned in a drawing and the dialog is re-launched to make changes.

Modify formatting Check to modify the formatting of areas so that areas in the same district appear in the same colors.

Create contiguous districts Use a networking algorithm to try to assign areas to districts so that areas in the same district are adjacent to each other. On by default. If this option is turned off, the system can work much faster but areas might be assigned to districts in such a way that not all areas in the same district are adjacent to each other.

OK Accept the current assignment and update the drawing with the districts as seen in the preview pane and reported in the districts list pane.

Cancel Exit the dialog without making any changes to the drawing.

Pre-Assigning Areas to Districts

To force an area to be assigned to a give district, highlight that district in the districts list pane and then click on the area. When the Assign button is pressed the district assignment will keep that area assigned to that district and will attempt to build the district around that area.

Example

Let’s consider an example using the sample drawing of Mexican provinces on the Manifold CD. We will assign different provinces to different districts.

When we open the drawing we see it rendered in default gray formatting. The drawing shows provinces in Mexico as areas.
If we open the drawing’s table we can see it has numerous fields. Visible in the illustration above is the ID field giving the internal object identifier, a SQMI field that gives the area of each province in square miles, a SQKM field that gives the area in square kilometers and a POBL_1990 field that gives the 1990 population of each province. There are more fields in the table outside the view shown in the illustration above.

With the focus on the drawing window, we choose **Drawing - Districts (Visual)** in the main menu.

The dialog launches as seen above. This illustration and other illustrations in this example that show the preview pane are shown in a reduced view to more easily fit into this documentation. The dialog has a visual preview pane at the top that shows the drawing, a districts list pane in the lower left that lists districts with information about each district, and a group of options to the right.

Looking more closely at the districts list pane we see it opens by default with two districts, colored yellow and blue, that at present have no areas assigned to them.

Looking more closely at the options we see that the dialog opens using the **Area (I)** intrinsic field as the balancing field by default. This is simply because the **Area (I)** intrinsic field is the first column in alphabetical order of the columns available in this drawing’s table.

We shall create districts based on the number of square miles of each province. To do so, we will change the **Balance** field to **SQMI**.
Click into the **Balance** box and choose **SQMI**.

The dialog is now set up to create two districts of approximately equal size, assigning areas so that total value of **SQMI** in each district is approximately the same. To assign districts, we press the **Assign** button. The system will compute for a while (potentially a long while with complex drawings that have many areas) and will assign areas to districts.

The illustration above shows a reduced view of the preview pane and part of the districts list pane.

We can see in the visual preview pane that each area has been colored yellow or blue depending the district to which it has been assigned. In the districts list pane we can see that 13 areas have been assigned to the yellow district, for a total **SQMI** value of 410588.96 and that 19 areas have been assigned to the blue district for a total **SQMI** value of 411531.37. Manifold helpfully tells us the total percentage of **SQMI** in both the yellow and the blue districts. We can see that with the yellow district having 49.94% and the blue district having 50.06% the assignment of areas to districts worked out very well for the objective of having about the same total **SQMI** in each district.
It’s not always possible, of course, to assign areas to districts so that the value being balanced works out so evenly. However, Manifold will do its best and will usually assign areas to districts with much better accuracy than can be done by hand. The task is especially difficult if the **Create contiguous districts** box is checked (the default) so that not only must districts be balanced but they must also have all areas in contiguous contact with each other.

We can try a different arrangement of districts. To do so, we first press the **Clear All** button to clear all assignments.

The visual preview pane will go back to the default unassigned gray for all areas and the districts list will clear all values for the two districts.

**Create contiguous districts**

We can next try creating districts again, but this time unchecking the **Create contiguous districts** box to see the effect of relaxing this constraint. Once more we press **Assign**. This time, the assignment process happens virtually instantaneously since relaxing the requirement to create contiguous districts allows Manifold to work much faster.
With the *Create contiguous districts* box turned off, the districts created are rather similar, but we can see from the districts list pane that the balance is not quite as good. In this case, the percentages are 47.95% and 52.05%, not quite as even as with the *Create contiguous districts* box checked on. This shows a secondary effect of this option box, in that it not only directs Manifold to form contiguous districts but as a side effect of the algorithm used to form contiguous districts it will often end up with slightly better balancing.

**Using Three Districts**

![Create contiguous districts checkbox]

Let's check the *Create contiguous districts* box once more, and let's add a third district and assign the provinces into three districts instead of only two. We press the **Clear All** to clear district assignments and make way for new assignments.

To add a third district, click the **New** button in the districts list pane toolbar.

This adds a third district to the districts list, by default colored red. If we would like to change the colors of any of the districts, we can double-click into the district's color well to change the color.

After adding the third district we press **Assign**.

Manifold will compute optimal assignments for three districts and assign each area to a red, yellow or blue district. We can see from the percentages that districts have been assigned quite evenly, with approximately 33% of the total SQMI area in each district.
How would the districts look if we unchecked the Create contiguous districts box?

We uncheck the Create contiguous districts box, press Clear All to clear district assignments and then press Assign.

The result is not quite as good as with the Create contiguous districts box checked. Note that the percentages are not as close to an even 33% for each district and also that one of the areas in the red district is not adjacent to the others. Note also that Manifold has switched the colors of districts assigned, with the blue district now occurring in the North.

In general, when we ask Manifold to assign districts from a blank slate the system has a free hand in how areas are assigned. If we would like to control the assignment of areas to districts, we can do so.

Manual Assignment of Districts

Let's begin by checking the Create contiguous districts box once more so contiguous districts are created. We will also press Clear All to start with a blank slate once more.
To assign an area to a district, click on the district in the districts list pane to highlight it and then click on any area in the preview pane. If an area is unassigned or has been assigned to a different district, clicking on it will assign it to the highlighted district. If an area has already been assigned to the highlighted district, clicking on it will clear the assignment. In the illustration above we have highlighted the red district in the districts list pane and have already clicked on two unassigned areas to assign them to the red district. The districts list shows statistics for two areas in the red district.

As areas are added to or removed from each district the districts list pane will show updated statistics. The districts list pane now shows statistics for three areas in the red district.

To assign areas to a different district, click on that district in the districts list pane to highlight it. In the illustration above we have clicked on the yellow district in the districts list pane and then we’ve clicked on three areas in the preview pane to add them to the yellow district. Note that the districts list pane has been updated with information on the three areas in the yellow district as well.
We can click on the blue district and then click on five areas in the preview pane to assign them to the blue district.

The areas that are assigned to a district will stay assigned if we now press the Assign button to assign the remaining areas to districts. Pre-assigning areas manually is a way of forcing them to be part of a desired district and to compel Manifold to do the best job it can of balancing the remaining areas into districts.

If we press the Assign button Manifold will balance the remaining districts as seen above. By pre-assigning some areas to the red, yellow and blue districts we can guide the desired shape of a district, perhaps at some cost to the optimization of balancing.

Re-balancing Districts

We can also clear some districts and force Manifold to re-balance them while retaining existing districts.
To clear a district, highlight the district in the districts list and then press the **Clear** button. In the illustration above we have highlighted the red district and will now **Clear** it.

The result is that all areas formerly in the red district have now been unassigned.

We can also highlight the yellow district and then press the **Clear** button to clear all areas in the yellow district as well.
If we press the **Assign** button Manifold will assign all unassigned areas to the three districts. Note that in making this assignment Manifold added two small areas to the blue district in addition to those areas that were already assigned to the blue district. When areas are pre-assigned Manifold will never remove them from the district but it is always possible that Manifold will add some areas to a district.

### Districting with Different Weights

So far this example has assigned areas to districts where each district had a weight of 1. This means that the objective has been to create districts that all have the same total value of the balancing field. If desired, we can change the weights on districts so that, for example, one district is created three times larger than the other districts.

We will now re-balance the districts so that the red district is three times larger than the yellow or blue districts.

We begin by pressing the **Clear All** button. Next, we double-click into the 1.00 weight in the red district's line and change it to 3.00.
When we press **Assign** Manifold will attempt to assign areas to districts so that the red district is three times larger than the yellow or blue districts. As we can see above, the assignment is remarkably close to the ideal. At almost 60% the red district is indeed three times larger than the other districts, which are approximately 20% each.

However, we might not be happy with the shape of the red district. Perhaps we would be willing to sacrifice some of the accuracy of the numeric balancing in order to get districts that are more compact. If desired, we can adjust the districts manually.

For example, we could highlight the blue district and then move two of the areas assigned to the red district into the blue district by clicking on them.

Next, we could highlight the yellow district and then click on one of the blue areas to move it into the yellow district. The net result of these changes is a more compact shape to our districts but at the cost of increasing the share of the smaller districts to about 25% each and decreasing the share of the larger district to about 53%.
Working with districts is often a tradeoff between having districts that are numerically close to an ideal value and having districts that are compact or otherwise have a desired shape. We rarely have the opportunity to assign districts that have exactly the shape as well as the numeric balance desired. Manifold’s Districts (Visual) dialog is therefore generally used interactively to change options, pre-assign some districts and otherwise develop the assignment desired.

Results

So far, all work has happened within the Districts (Visual) dialog. If we press OK, the formatting will be transferred to the drawing and the drawing will be updated with a new Districts field that contains the number of each district for each area.

When we press OK the drawing is updated with the formatting used in the dialog.

<table>
<thead>
<tr>
<th>Mexico Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
</tr>
<tr>
<td>-----</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
</tbody>
</table>

If we open the drawing's table, we will not initially see a District field because it is hidden by default.

To make it visible we choose the Table - Columns dialog and click the District field on. In the illustration above we have also used the Move Up button to move this field near the top of the table.
The **District** field will then appear. We can see that each area has been assigned to one of districts 1, 2 or 3. If an area was not yet assigned it would have a value of 0 in the **District** field.

### Using the Load Districts Button

If a drawing's table already has a districts field (which could be called **District** or some other name) with district assignments for each area we can load those assignments into the dialog by choosing the field in the **Save to** box and then pressing the **Load Districts** button.

### Performance

Assigning areas to districts can be computationally very intensive if the **Create contiguous districts** box is checked, as we would normally want it to be. It is possible to redistrict a drawing with approximately a thousand areas in less than fifteen minutes on modern machines with adequate memory, however larger drawings can take longer.

### Note

This example has balanced districts using the **SQMI** field in the drawing, which gives the area of each province in square miles. We've used this field because it makes it immediately visually obvious what is being balanced, the total size of each district. However, districts are more frequently balanced using some other field, such as total population or total number of customers, that does not have a direct relationship with the surface area of the areas being assigned.

### Tech Tip

In the illustration above, we have created districts that are balanced using the **POBL_1990** field that gives the population for each province. Each district has approximately the same population, remarkably close to 33.33% for each district. The districts have unequal geographic area because much of Mexico's population is concentrated within densely populated provinces near Mexico City.
It is frequently the case that we would like to give districts names such as "North," "Central," and "South" instead of using numbers. There are two ways of approaching this. The first is to change the type of the District field to a text type and then in the table select all records with District equal to 1, change them to "North" (easy to do when they are all selected because changing one changes them all) and so on. However, that loses the formatting and district information from the numeric District field, since the Districts (Visual) dialog works only with numeric fields. If we take this approach we will have to manually format the area colors using a thematic format based on unique text values of the District field.

Another approach is to create a new text column in the table, perhaps called District Name or some other convenient name and to then use the transform toolbar to copy the values from the District column to the District Name column. Manifold will automatically translate type on the fly from numeric to text. We can then select all records with District Name equal to 1, change them to North and so on. This will retain the original numeric District field and formatting while also providing us with a text field that can be used for labels and so on.

See the Editing Data in Tables topic for information on selecting cells in records and changing all selected items at once.

See Also

Districts (Advanced)

Districts
Drive-Time Zones

The **Drawing - Drive-Time Zones** menu selection is enabled when the optional Business Tools extension has been installed and the focus is on a drawing window or on a drawing layer in a map window that contains lines and at least one point (which can be used as a center). If you have not activated the Business Tools extension with a valid Business Tools serial number you will not be able to use the Drive-Time Zones command.

Manifold's Drive-Time Zones command can find zones from one or more points, called Centers from which travel occurs. A drive-time zone is the distance that can be traveled through the road network (and possibly off road as well) in the given time.

The drawing of road lines used must be a true road network that is a connected network. If there are small breaks between the road lines or if the ends of adjacent lines are not exactly coincident the drawing cannot be used as a road network. The drawing used must be **projected** so that accurate measurements can be made.

**Controls**

**Center** Choose a set of points (usually the [Selection] or a saved selection) that are to be used as centers for the drive time zones. **Very Important:** Points to be used as centers must be exactly coincident with the end of one of the lines in the road network.

**Roads** Choose a set of lines to be used as a road network. Usually [All Objects] to use all lines in the drawing, or might be a saved selection of lines to use only some lines from a very large drawing if we know that the drive time zones will end up being reasonably local.

**Length** Choose a field to use that gives the length of each line in the road network. Choose the unit of measure used for the value in the field. Use the intrinsic field Length (I) if there is no explicit length field. On large road networks for which drive-time zones may be repeatedly calculated, it is more efficient to create a "length" field in the table and to copy into that length column the contents of the Length (I) intrinsic field using the transform toolbar for tables. This will avoid a recalculation of Length (I) on the fly for each run and will improve performance.

**Speed** Choose a field to use that gives the speed of travel for each line in the road network. Choose the units of measure used for the length and time values in the field, for example, miles per hour or kilometers per hour. By default, if a column named Speed is available in the drawing's table it will be automatically loaded into the Speed box.

**Split roads at intersections** If checked, split road lines into separate lines at intersections with other roads.

**Method** Choose **Buffer**, **Hull** (default) or **Zone** as follows:

**Buffer** - Find all portions of roads that may be reached in the given time and then build a buffer zone area about each road line that extends out the given distance. Specify the distance desired and the units to be used (meters, feet, etc.) for the distance. **Use with care**, since phenomenally complex buffer zones can be created in urban areas that may take a long time to compute.

**Hull** - Find the most distant spot on each road that may be reached in the given time and create an area that is the convex hull enclosing the most distant points. This is the usual "drive time zone" polygon created by ordinary GIS systems.

**Zone** - Find the most distant spot on each road that may be reached in the given time as well as the additional area that may be reached by travelling from each reachable spot at the given Off-road speed. The Off-road speed value is specified using whatever units of
linear measure are used in the coordinate system of the projected drawing and whatever units of time measure are used for the Speed combo. For example, if the drawing is in a meter-based projection like Orthographic and the Speed value is in miles per hour the value for Off-road speed will be in meters per hour. The Zone method serves the classic task of trying to find how far someone can get in a given time if they travel either/or by road or by leaving the road and travelling over the ground off-road (often used for law enforcement search planning).

**Zones**
The drive time for each zone and the units of measure (hours, minutes, etc.). For example, to create a single drive time zone enter 60 for the distance reachable in that time. Enter 60, 120, 180 to create three drive time zones each of which is reachable in times of 60, 120 and 180. **Note:** this requires the Tools - Options - Miscellaneous - Use English measurement units option to be set the same as the linear units of measure used by the coordinate system.

**To create a drive time zone from a single center:**

1. Open a projected drawing that contains lines that form a connected road network and at least one point. Lines should have a field that can be used as the Speed field. If all roads are to be treated as having the same speed, simply create a “speed” field and fill it with 1 for all lines.
2. Select a point on the road network (lying exactly at the end of one of the lines that make up the network) that will be the center of the drive-time zone. Launch the Drive-Time Zones command.
3. Choose [Selection] for the Center value.
5. Choose Length (I) to use as the Length value as well as the unit of measure.
6. Choose the name of the field giving the speed value in the Speed combo as well as the units of measure.
7. Choose the desired Method.
8. In the Zones field, choose the time desired for each zone plus the unit of time measurement. Press OK.

When choosing Method, it is a good idea to start with a simple method, such as Hull, to see just how much of a region will be covered by a given set of parameters in a drive-time zone. This will reveal situations where settings accidentally result in very large drive-time zones that may cover many streets in an urban area. In such cases, use of the Buffer method may create extremely complex buffer zone areas that can take a long time (many hours) to compute. It is a good idea to test the desired parameters using a rapidly computed method such as Hull before launching a more compute-intensive method such as Buffer.

**Example**

We begin with a drawing of streets that has several points, one of which has been selected. The drawing is in an Orthographic meter-based projection so that distances are in meters and may be measured accurately.
The streets (as usual) consist of many lines with the end points of the lines coincident to form a connected network, that is a real network without any gaps or overshoots between the lines. The points are located exactly at the ends of lines so that they two are part of the network.

<table>
<thead>
<tr>
<th>ID</th>
<th>FENAME</th>
<th>CFCC</th>
<th>Length (I)</th>
<th>Length</th>
<th>Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>60303</td>
<td>High</td>
<td>A1</td>
<td>476.9169...</td>
<td>476.916...</td>
<td>10</td>
</tr>
<tr>
<td>60304</td>
<td>Everett</td>
<td>A1</td>
<td>262.74861...</td>
<td>262.748...</td>
<td>10</td>
</tr>
<tr>
<td>60305</td>
<td>University</td>
<td>A1</td>
<td>46.63075...</td>
<td>46.630...</td>
<td>45</td>
</tr>
<tr>
<td>60306</td>
<td>Mitchell</td>
<td>A1</td>
<td>476.92415...</td>
<td>476.924...</td>
<td>10</td>
</tr>
<tr>
<td>60307</td>
<td>University</td>
<td>A1</td>
<td>123.96614...</td>
<td>123.966...</td>
<td>45</td>
</tr>
<tr>
<td>60308</td>
<td>Ama</td>
<td>A1</td>
<td>186.52893...</td>
<td>186.528...</td>
<td>85</td>
</tr>
</tbody>
</table>

If we open the drawing’s table we see that most lines have a speed of 10 marked on them with some lines having higher speeds. In the drawing, lines with faster speeds have been thematically formatted so they are in different colors. The drawing’s table also has the Length (I) intrinsic field showing as well as a Length field that we created and populated by copying the contents of the Length (I) intrinsic field. We could use the Length field if we had a really big network and wanted the calculations to go slightly faster, but it is not used in this example.

Launching the Drive-Time Zones command with the above settings will create a drive-time polygon in the shape of a convex hull about the selected point that was used as the Center.

Note that the units used are mixed, with meters, miles per hour and minutes in use. The Length parameter is in meters, since the use of Orthographic projection cast the native dimensions in use in this drawing in meters. The units used for Speed are simply whatever is intended by the values in that column in the drawing. The example drawing is of a location in the United States, where speeds are recorded in miles per hour.
Manifold simply reaches out from the center point as far as is possible to go through the road network and then draws a polygon to enclose the most distant spots reachable. Because the two colored roads have much faster speeds than the smaller streets around them, by far the greatest reachable distance is over the faster roads. The maximum distance reachable in the given time over the faster roads thus marks the corners of the resultant drive-time zone.

If we delete the convex hull drive-time zone area and re-run the command using the same central point but using the **Buffer** option, we can create a drive-time buffer using a setting of **30** meters.

The result is more interesting than using the **Hull** option. What we see is how far we can get from the center point along all of the roads in the given time, plus a buffer zone that extends out 30 meters from each reachable spot. Note that because of the great difference in speed between the main roads and the smaller streets, we can get only a small distance down any of the side streets in the given time.
The **Buffer** option is often used to find which addresses can be reached in a given time from a central location. Addresses are often placed in drawings using the geocoder and so they are often not exactly on the street line but close to the street line. By creating a buffer zone as a drive-time zone, we can find which address dots are located within the resultant buffer zone area.

Using the **Buffer** option provides a very rich display, but it can be very computationally intensive. In a road system with many roads, even a slight increase in the drive time distance can result in a considerably more complex buffer area with a surprisingly large increase in computation time required. Use this option with great care, scaling up in small steps to avoid launching Manifold on a job that could take days to compute.

A third option is to use the **Zone** option as seen in the dialog above, using an **Off-road** speed of 3200 to represent 3200 meters per hour, approximately 2 miles per hour. The linear units for the **Zone** speed are taken from the projection, in this case **meters** from the Orthographic projection and the time units of measure are the same as are specified for the **Speed** parameter, in this case being **hours** from the miles per hour speed values.

The resultant drive-time zone is very interesting. It shows how far one can travel from the central point using a combination of travel on the roads and travel off the road. For regions far from the central point the zone shows those spots reachable when travelling all the way or almost all the way by car at the given road speed. That's why the zone is elongated along the lengths of fast main roads.

Closer in to the central point the zone covers those spots that are reachable if we go a short way by the road network and then get out of the car and travel off-road over the ground at the given off-road speed for whatever time is left. Therefore, close to the central point the regions between roads get filled in because they can be reached within the given time by off-road travel.
If we had re-run the command using an off-road speed of 5 we would have obtained the result above. It shows that if our off-road speed is faster, we can get to even more regions in between roads.

The Zone option is often used by law enforcement or search and rescue services to define an area to search when a period of time has elapsed and the target person may be travelling on or off roads. For example, suppose some people in a four-wheel drive vehicle are lost in a semi-desert region where it is not clear if they stayed on the road during their travel or ventured off-road into the desert. If we know where they started and how much time has elapsed we can make reasonable guesses at their speed on the road network and their speed off-road. We can then use the Zone option to find the region to search, consisting of that region that they could reach through any combination of on-road and off-road travel.

**Troubleshooting**

Simple errors that may cause problems:

- Points to be used as locations are either not on lines or are not within the drawing's precision factor distance of a line. Since the default precision factor for a projected drawing is 0.000001 meter (about one thousandth of a millimeter) it is highly unlikely that points placed in drawings using anything other than Snap functions will be adequately close to a line to be considered "on" the line.
- The drawing of roads might not be a real network, that is, the ends of adjacent lines may not be coincident, there may be dangles or undershoots, breaks in the lines or other disconnects that prevent the lines shown from being used as a real network. Use Normalize Topology to "clean" the road network if need be, but keep in mind that poorly drawn road drawings might not be repairable even with the Normalize Topology tool.
- The units of measure used in the dialog do not correspond with the units of measure used in the drawing or used in the Speed or Length fields. For example, if a drawing has been projected using a foot-based projection the use of meters to interpret a Length (I) intrinsic field will be inaccurate.
- The drawing is not projected and covers a sufficiently large area that errors arise. Always use a drawing that has been projected using a projection appropriate to the region of interest. For small regions (say, the size of a state or province in most countries), Orthographic is a fine meter-based projection that is a good bet.
- Changing the setting of requires the Tools - Options - Miscellaneous - Use English measurement units option to be different from the linear units of measure used by the projection / coordinate system will cause inaccurate results. If the projection is in meters, that option should be unchecked. If the projection is in feet, the option should be checked. It is possible to leave the option in an incompatible state and to multiply the Zones values by the ratio feet to meters or meters to feet and still get an accurate result, but that is difficult to keep in mind and not recommended.

**Note**

Strictly speaking, a drawing used for drive-time zone computation need not be projected. However, since degrees of Latitude vary in size from the Equator to the Poles any task that requires accuracy should use only projected drawings.

**See Also**

Network User's Introduction
Optimal Route

The Drawing - Optimal Route menu selection is enabled when the optional Business Tools extension has been installed and the focus is on a drawing window or on a drawing layer in a map window that contains lines and at least two points (between which a route can be selected). If you have not activated the Business Tools extension with a valid Business Tools serial number you will not be able to use the Optimal Route command.

The Optimal Route tool is a less-interactive version of the Optimal Route (Visual) tool. It simply finds the best route through a given set of locations (the selection or a saved selection), taking the locations in free order. To learn how to use this tool, see the Optimal Route (Visual) topic.

The drawing of road lines used must be a true road network that is a connected network. If there are small breaks between the road lines or if the ends of adjacent lines are not exactly coincident the drawing cannot be used as a road network.

The Optimal Route tool is provided for expert use as a specialized shortcut. In almost all cases for interactive work it is better to use the Optimal Route (Visual) dialog. However, the parameters used in Optimal Route are directly analogous to those used in the programming model when scripting optimal routes. The Optimal Route dialog is therefore especially useful to programmers who are learning to work with Manifold's optimal route capabilities via the scripting interface.

Controls

- **Locations**: Points to use for locations. [All Objects] or [Selection] or a saved selection.
- **Roads**: Objects to be used for road lines. [All Objects] or [Selection] or a saved selection.
- **Length**: Choose a field to use that gives the length of each line in the road network. Choose the unit of measure used for the value in the field. Use the intrinsic field Length (I) if there is no explicit length field. On large road networks for which optimal routes may be repeatedly calculated, it is more efficient to create a "length" field in the table and to copy into that length column the contents of the Length (I) intrinsic field using the transform toolbar for tables. This will avoid a recalculation of Length (I) on the fly for each run and will improve performance. Lines with a zero or negative value for Length will be ignored and will not be used in the route.
- **Speed**: Choose a field to use that gives the speed of travel for each line in the road network. Choose the units of measure used for the length and time values in the field, for example, miles per hour or kilometers per hour. By default, if a column named Speed is available in the drawing's table it will be automatically loaded into the Speed box.
- **Split roads at intersections**: If checked, split road lines into separate lines at intersections with other roads.
- **Save report**: Check to create a comments component in the project that contains a driving directions report.
- **Names**: Field to use for road names in the driving directions report. If [None] is selected, the report will be created using a format without road names. By default, if a column named Name is available in the drawing's table it will be automatically loaded into the Name box.
- **Turns**: Choose the writing style used to report turns. none results in no turns reported. The other two styles allow choice of turns using either compass directions (for example, to turn North or South at a given intersection) or relative directions (for example, to turn right or left at a given intersection).
OK  Find and optimal route and add a line object to the
drawing representing the route. Add a comments
component with driving directions if the Save report box
has been checked. If no route has been as yet created
using the Locate Route button, pressing OK is the same
as Cancel.

Cancel  Exit the dialog without making any changes to the
drawing.

Units of Measure used by Intrinsic Fields

The units of measure used by the Length (I) and Area (I) intrinsic fields depend upon both the projection of the
drawing and the Tools - Options setting for the Use English measurement units option as follows:

- If the drawing is in Latitude / Longitude (that is, unprojected) Length (I) is reported in degrees and
  Area (I) is reported in square degrees.
- If the drawing is projected and the Use English measurement units option is turned off, Length (I) is
  reported in meters and Area (I) is reported in square meters.
- If the drawing is projected and the Use English measurement units option is turned on, Length (I) is
  reported in feet and Area (I) is reported in square feet.

Troubleshooting

Simple errors that may cause problems:

- Points to be used as locations are either not on lines or are not within the drawing's precision factor
distance of a line. Since the default precision factor for a projected drawing is 0.000001 meter (about
one thousandth of a millimeter) it is highly unlikely that points placed in drawings using anything other
than Snap functions will be adequately close to a line to be considered "on" the line.
- The drawing of roads might not be a real network, that is, the ends of adjacent lines may not be
  coincident, there may be dangles or undershoots, breaks in the lines or other disconnects that prevent
  the lines shown from being used as a real network. Use Normalize Topology to "clean" the road
  network if need be, but keep in mind that poorly drawn road drawings might not be repairable even with
  the Normalize Topology tool.
- The units of measure used in the dialog do not correspond with the units of measure used in the
  drawing or used in the Speed or Length fields.
- The drawing is not projected and covers a sufficiently large area that errors arise. Always use a
drawing that has been projected using a projection appropriate to the region of interest. For small
regions (say, the size of a state or province in most countries), Orthographic is a fine meter-based
projection that is a good bet.

See Also

Network User’s Introduction
Optimal Route (Visual)
Optimal Route (Visual)

The **Drawing - Optimal Route (Visual)** menu selection is enabled when the optional Business Tools extension has been installed and the focus is on a drawing window or on a drawing layer in a map window that contains lines and at least two points (between which a route can be selected). If you have not activated the Business Tools extension with a valid Business Tools serial number you will not be able to use the **Optimal Route (Visual)** command.

The **Optimal Route (Visual)** command provides an interactive dialog that allows choosing locations within a road network so that an optimal route between the locations may be found through the road network, taking into account the length of the roads and speed limits on the roads. In the main display pane the dialog shows a visual preview of the drawing in which locations are selected and routes are found. The dialog also lists the locations to be traversed and relevant options for the road network.

Locations to be visited may be selected from any point in the drawing meeting one of the following criteria:

- The point is located exactly at the endpoint of a line.
- The point is located exactly at one of the coordinates that defines a line.
- The point is located closer to the line than the distance given in the drawing’s location precision property (open the drawing, click on View - Properties and then the Precision […] button).

The drawing of road lines used must be a true road network that is a connected network. If there are small breaks between the road lines or if the ends of adjacent lines are not exactly coincident the drawing cannot be used as a road network.

To operate the dialog, open a drawing that has a road network of lines as well as points on the lines that are to be visited. Launch **Optimal Route (Visual)** and then in the preview pane click on the points to be visited to add each point as a location. As each point is clicked, it will be highlighted with red selection color and added to the list of locations to be visited in the lower left pane of the dialog. The start location will be shown with a larger red dot and the end location will be shown with a larger red diamond. Clicking a point that has been already selected will deselect it from the route.

By default, the system will find an optimal path through all of the locations from the first location to the last location regardless of the order in which locations between the first and the last appear in the list. To force the system to find an optimal path through the locations in the order in which they appear in the locations list, check the **Use list order** box before locating a route. To order the locations within the list, either click them in the preview pane in the desired order or change the order in the list. To re-order locations in the list use the up and down buttons in the locations list pane’s toolbar.

If any points are selected in the drawing when the **Optimal Route (Visual)** dialog is launched they will be automatically added to the locations list for the route. A start location may then be designated by highlighting it in the locations list and clicking the **Move to Top** button and an end location may be designated by highlighting it in the locations list and then clicking the **Move to Bottom** button. If the dialog is re-run in a drawing and no points are selected, the dialog will launch with whatever was the last used list of locations. This makes it easy to re-run the dialog without having to re-click points into the locations list.

Locations will be listed in the locations list pane using their object ID numbers. If each point has a field called “Name” pressing the **Show Name** button will show the contents of the Name field for each location in the list. Using the **Show Name** button provides a much more user friendly list of locations than just using the object ID's.

Options in the lower right section of the dialog allow choosing which field specifies the length of the roads, the field specifying the speed of travel over each road and the units of measure. If desired, a report can be created using a variety of styles that provides driving directions for the route.

When all options have been selected, pressing the **Locate Route** button will identify the optimal route between the points to be visited. The route will be shown in red selection color. Pressing the **Trace Route** button will launch an animation that shows a small dot travelling over the route. When a route has been located, we can click the **Save Report** button to save a report of driving directions as a comments component in the project under whatever name we like.

When happy with the located route, press **OK** and the route will appear in the drawing as new line object that traverses the route. The new line object will be selected. To avoid creating duplicated objects in the drawing it is a good idea to **Cut** the new route line object and to **Paste** it into a new drawing.
To find an optimal route between locations:

1. Open a projected drawing that contains lines that form a connected road network and at least two points. Lines should have a field that can be used as the Speed field. If all roads are to be treated as having the same speed, simply create a "speed" field and fill it with 1 for all lines. Launch the Optimal Route (Visual) dialog.

2. In the preview pane, click on the point that is to be the starting point for the route. Next, click on each point that is to be visited in order. All points must be on the road network (that is, lying exactly at the end of one of the lines that make up the network). The clicked points will appear as locations to be visited in the locations list.


4. Choose Length (l) to use as the Length value as well as the unit of measure. The unit of measure will depend on the projection in use and is normally meters (for meter-based projections like Orthographic) or feet (usually used only in US specific projections).

5. Choose the name of the field giving the speed value in the Speed combo as well as the units of measure (the default is meters per hour, which quite likely will be changed to miles per hour or kilometers per hour).

6. Press the Locate Route button to see a preview of the route in the preview pane. To animate the route, press the Trace Route button.

7. If a driving instructions report is required, choose the report options desired and click the Save report button. This will create a new comments component in the project that reports the driving instructions for the route.

8. If happy with the route, press OK. A new line object will be created in the main drawing and highlighted in red selection color. The line object is a single line that includes all coordinates traversed in the route. To save this line, Cut it, create a new drawing and then Paste it into the new drawing. It is usually useful to save the new line that represents the route so that it may be used together with the original drawing in a map, with the new route line formatted to make the route obvious (using a thicker line, etc).

9. If the Save report option was chosen, the system will generate a report and save it into a new comments component.

Controls

(Visual preview pane) Displays the drawing in default gray color.

- **Zoom To Fit** - Zoom so that the entire drawing fits within the preview pane.
- **Zoom In** - Magnify the view as if seen from a closer distance.
- **Zoom Out** - Reduce the view as if seen from farther away.
- **Zoom Box** - Zoom to the size of the cursor box drawn with the mouse.
- **Center Point** - Pan the view so that the spot clicked is centered.
- **Grabber** - Interactive pan: click and drag with the grabber hand. The scene will be panned so that the initial point is moved to the spot where the drag is released.
- **Trace Route** - Animate the route using a moving black dot to trace out the route.

(locations list pane) A list of locations to be visited in order by the route. The starting location is at the top of the list and the end location is at the bottom of the list. Location are identified by their internal Manifold ID numbers and, if desired, by the contents of a Name field in the drawing's table if the Show Names button is pushed in.

- **Delete** - Delete the highlighted location from the locations list pane. Clicking a location that has been already selected will also deselect it from the route and delete it.
from the locations list pane.

- **Center** - Center the highlighted location in the window.

- **Ping** - Locate the highlighted location by drawing converging circles to it. A great way to see what the highlighted location is.

- **Move to Top** - Move the highlighted location to the top of the locations list pane.

- **Move Up** - Move the highlighted location up one position in the locations list pane.

- **Move Down** - Move the highlighted location down one position in the locations list pane.

- **Move to Bottom** - Move the highlighted location to the bottom of the locations list pane.

- **Show Names** - In the locations list display names for locations taken from a Name column for the point in the drawing's table.

**Use list order**

If checked, Manifold will create an optimal route that goes through all of the locations in the locations list in order. If not checked (the default) Manifold will begin at the first location and end up at the last location but will find an optimal route that goes through all of the other locations at least once in the best possible way.

**Locate Route**

Enabled when two or more points have been selected in the preview pane and added to the locations list. Press to create an optimal route between the locations in the desired order. **Note:** until the OK button is pressed, the located route is only a preview.

**Roads**

Objects to be used for road lines. [All Objects] or [Selection] or a saved selection.

**Length**

Choose a field to use that gives the length of each line in the road network. Choose the unit of measure used for the value in the field. Use the intrinsic field Length (l) if there is no explicit length field. On large road networks for which optimal routes may be repeatedly calculated, it is more efficient to create a "length" field in the table and to copy into that length column the contents of the Length (l) intrinsic field using the transform toolbar for tables. This will avoid a recalculation of Length (l) on the fly for each run and will improve performance. Lines with a zero or negative value for Length will be ignored and will not be used in the route.

**Speed**

Choose a field to use that gives the speed of travel for each line in the road network. Choose the units of measure used for the length and time values in the field, for example, miles per hour or kilometers per hour. By default, if a column named Speed is available in the drawing's table it will be automatically loaded into the Speed box.

**Split roads at intersections**

If checked, split road lines into separate lines at intersections with other roads.

**Save report**

Enabled when a route has been located. Press to create a comments component in the project that contains a driving directions report.

**Names**

Field to use for road names and location names in the driving directions report. If [None] is selected, the report will be created using a format without road names or
location names. When names are used, the report will note all points that are passed by the route even if they are not in the locations list. By default, if a column named Name is available in the drawing’s table it will be automatically loaded into the Name box.

**Turns** Choose the writing style used to report turns. none results in no turns reported. The other three styles allow choice of turns using either compass directions (for example, to turn North or South at a given intersection) or relative directions (for example, to turn right or left at a given intersection), of both relative and compass (the default).

**OK** Accept the current route shown in the preview pane and add a line object to the drawing representing the route. If no route has been as yet created using the Locate Route button, pressing OK is the same as Cancel.

**Cancel** Exit the dialog without making any changes to the drawing.

**Example**

Let's use the Optimal Route (Visual) dialog to find a route between locations.

We begin with a drawing of roads that has several points in the drawing as well. The points have been exactly located at the ends of lines using snap mode to snap to the ends of lines when creating points.

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>CFCC</th>
<th>Length (ft)</th>
<th>Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>60729</td>
<td>Ramona</td>
<td>A41</td>
<td>431.77023...</td>
<td>10</td>
</tr>
<tr>
<td>60730</td>
<td>Bryant</td>
<td>A41</td>
<td>454.34580...</td>
<td>10</td>
</tr>
<tr>
<td>60731</td>
<td>University</td>
<td>A41</td>
<td>279.79527...</td>
<td>45</td>
</tr>
<tr>
<td>60732</td>
<td>El Camino</td>
<td>A31</td>
<td>277.99892...</td>
<td>70</td>
</tr>
<tr>
<td>60733</td>
<td>Aina</td>
<td>A41</td>
<td>455.47210...</td>
<td>85</td>
</tr>
<tr>
<td>5074</td>
<td>Lane 13</td>
<td>A41</td>
<td>262.74813...</td>
<td>10</td>
</tr>
</tbody>
</table>

If we open the drawing's table we can see that each line has a Name and a Speed for the road represented by that line. The speed shows the actual speed used on that road which in some cases is substantially larger than the posted speed limit. We have the Length (ft) intrinsic field turned on for display so that at a glance we can see the length of each road line.
If we select the points and view the drawing's table with the Selection Filter button pushed in so that only the selection is visible, we can see that the each point also has a Name value describing the point.

When we launch the Optimal Route (Visual) dialog it shows the drawing in a preview display in the upper pane. Points in the drawing are shown in bright green color by default. Any color desired may be specified in the Tools - Options - Color page.

We begin building a route by clicking the start location in the dialog's preview pane.
When we click the first location desired, it appears as a larger red diamond. The red color indicates it has been selected as part of the route and the larger diamond indicates it is the end location of the route (with only one location selected the initial location is also the end location).

In the locations list pane a new location is listed using the Manifold object ID number.

If we press in the Show Names mode button the locations list pane will report the Name for that location as well.

To add the next location we click on it in the preview pane.
When clicked it also appears as a larger red diamond. The larger diamond indicates it is the end location for the route. The initial location has changed to a larger red dot. The larger dot indicates it is the start location for the route.

In the locations list pane the new location appears below the first location.

We can add a third location by clicking on it in the preview pane.

The last location added appears as a larger red diamond to indicate it is the end location. Note that the previous end location (now the middle location of the route) appears as a smaller red dot, the red color indicating it has been selected as part of the route.
The third location added to the route also appears at the bottom of the locations list.

We can continue in this way to add two more locations to the route. The start location and the end location are shown as a larger dot and a larger diamond respectively and the intermediate locations appear as smaller red dots.

Each location also appears in the locations list.

Before finding routes we set up options. For the length of the road lines we will use the Length (I) intrinsic field. Since the drawing is in Orthographic projection and Use English measurement units is turned off in our installation, we use meters as the unit of length. The speeds are given in miles per hour using the Speed field in the table. For the report, we will use the absolute style of directions, using compass directions.
If we press the **Locate Route** button a route will appear in the preview pane. If we press the **Trace Route** button we can see how the route passes through the various locations from start location to end location. Some of the road segments are used more than once.

We can also press the **Save report** box so that a comments component giving driving directions will be generated.

If we press **OK** to accept the route, a new line object is generated in the drawing and a comments component containing driving instructions is added to the project. The new line, shown in red selection color, might be difficult to see in the drawing. We use **CTRL-X** or **Edit - Cut** to **Cut** the new line object out of the drawing. We then paste it into the project pane as a new drawing.
We can then show the new drawing in a map together with the original roads drawing as seen above. We've formatted the new drawing with thicker lines so that the red selected route line is more visible.

Start at Fire Station.
Turn South-East on Middlefield.
  Drive for 2 minutes (2.30 mi).
Arrive Park.
Continue South-East on Middlefield.
  Drive for 1 minute (1.58 mi).
Turn North-East to California.
  Drive for 3 minutes (0.54 mi).
Arrive Middle School.
Turn South-West on California.
  Drive for 3 minutes (0.54 mi).
Turn North-West to Middlefield.
  Drive for 1 minute (1.58 mi).

The driving directions (fragment shown above) enumerate each segment of the route in text form. The driving directions also show the total time, total distance and average speed over the route. The illustration above shows driving directions using the "absolute" Turns style.

Start at Fire Station.
Take Middlefield.
  Drive for 2 minutes (2.30 mi).
Arrive Park.
Keep straight on Middlefield.
  Drive for 1 minute (1.58 mi).
Turn left to California.
  Drive for 3 minutes (0.54 mi).
Arrive Middle School.
Make U-turn on California.
  Drive for 3 minutes (0.54 mi).
Turn right to Middlefield.
  Drive for 1 minute (1.58 mi).

The illustration above shows driving directions using the "relative" Turns style.

Start at Fire Station.
Take Middlefield.
  Drive for 2 minutes (2.30 mi).
Arrive Park.
Keep straight on Middlefield (South-)
  Drive for 1 minute (1.58 mi).
Turn left to California (North-East)
  Drive for 3 minutes (0.54 mi).
Arrive Middle School.
Make U-turn on California (South-West)
  Drive for 3 minutes (0.54 mi).
Turn right to Middlefield (North-West)
  Drive for 1 minute (1.58 mi).

The third illustration above shows the default style, which uses both compass absolute and relative turns driving directions.
Which style is desired is a matter of taste. The absolute style uses compass directions and so is less ambiguous than the relative style in case of errors where the driver gets off a route and must rejoin it at some later point. However, the relative style is easier for most ordinary people to follow when driving. The both style is the default because it is completely unambiguous.

It is often the case that a road shown in a drawing is made up of multiple lines laid end to end. Manifold will collapse the driving directions for such end-to-end lines of identical name into a single line in the report. However, if the road changes name along the end-to-end sequence of lines the driving directions will note the newly named road as a new line in the report.

**Units of Measure used by Intrinsic Fields**

The units of measure used by the Length (I) and Area (I) intrinsic fields depend upon both the projection of the drawing and the Tools - Options setting for the Use English measurement units option as follows:

- If the drawing is in Latitude / Longitude (that is, unprojected) Length (I) is reported in degrees and Area (I) is reported in square degrees.
- If the drawing is projected and the Use English measurement units option is turned off, Length (I) is reported in meters and Area (I) is reported in square meters.
- If the drawing is projected and the Use English measurement units option is turned on, Length (I) is reported in feet and Area (I) is reported in square feet.

**Troubleshooting**

Simple errors that may cause problems:

- Points to be used as locations are either not on lines or are not within the drawing's precision factor distance of a line. Since the default precision factor for a projected drawing is 0.000001 meter (about one thousandth of a millimeter) it is highly unlikely that points placed in drawings using anything other than Snap functions will be adequately close to a line to be considered "on" the line.
- The drawing of roads might not be a real network, that is, the ends of adjacent lines may not be coincident, there may be dangles or undershoots, breaks in the lines or other disconnects that prevent the lines shown from being used as a real network. Use Normalize Topology to "clean" the road network if need be, but keep in mind that poorly drawn road drawings might not be repairable even with the Normalize Topology tool.
- The units of measure used in the dialog do not correspond with the units of measure used in the drawing or used in the Speed or Length fields.
- The drawing is not projected and covers a sufficiently large area that errors arise. Always use a drawing that has been projected using a projection appropriate to the region of interest. For small regions (say, the size of a state or province in most countries), Orthographic is a fine meter-based projection that is a good bet.

**Notes**

Strictly speaking, a drawing used for optimal route computation need not be projected. However, since degrees of Latitude vary in size from the Equator to the Poles any task that requires accuracy should use only projected drawings.

When showing route lines in maps, using a line style with directional arrows is an easy way to show direction.

What do we do if our drawing is in Orthographic or other meter-based projection but we would like to have length units in miles? We can turn on the Use English measurement units option and then create an Active Column that converts the Length (I) intrinsic column from feet to the equivalent distance in miles and then use the Active Column in the Length box.

**Tech Tip on Performance**

The task of locating an optimal route through a given number of locations in optimal order is NP-full. That is, solving the task can be mathematically shown to require examination of all possible permutations of the source data to achieve an absolute optimum. Therefore, the task is not solvable for any number of locations above 10 to12 because the required computing time expands so enormously beyond that number of locations.
A computation strategy for locating optimal routes when there are more locations is to use an heuristic algorithm that gives good results even if the results are not a mathematically perfect optimal route. Manifold uses this strategy. When there are ten or fewer locations Manifold will compute an exact solution. With eleven or more locations, Manifold uses an heuristic algorithm to find the optimal route.

It sometimes comes as a surprise to network newbies that finding an optimal route through a number of locations is much harder if the system must find the best route through them when they are taken in any order than if locations are visited in a given list order. The reason is that finding an optimal route through many locations in a given order is a simple matter of repeatedly finding the shortest path between only two locations: from the first location to the second, from the second to the third and so on.

In contrast, finding a route that minimizes travel time through many locations in free order requires solving the entire problem before even the second location to visit can be determined.

See Also

Network User's Introduction
Optimal Route
Send Email

The Drawing - Send Email menu selection is enabled when the optional Business Tools extension has been installed and the focus is on a drawing window or on a drawing layer in a map window that contains points, lines or areas. The command is also available when the focus is on a table window. If you have not activated the Business Tools extension with a valid Business Tools serial number you will not be able to use the Send Email command.

Manifold's Send Email command adds direct email capability to Manifold System. If you have data that contains an email address in some data attribute field, you can automatically email to objects using those email addresses. Send Email works by sending email through the Windows MAPI email layer. If applications that also use MAPI such as Outlook or Outlook Express have been installed the email will end up being routed through the Outlook or Outlook Express outbox.

Using a computer that already has Microsoft Outlook or Outlook Express installed is recommended because Outlook provides a simple means of checking that your system has been configured to correctly send email. Before attempting to operate Send Email make sure you can send email from Outlook. If you can't send email with Outlook, please fix that problem first.

Controls

From Enter an email to appear as the "from" email. When emailing on a system that has Outlook installed, whatever is the default email sending account for Outlook will appear in the "from" header in the emails regardless what is entered into this From field.

To Choose the field that contains email addresses. For example, if we have a drawing of points representing customers where the drawing's table has an "email" field that contains each customer's email address, we would choose the email field.

Email addresses must be in the form username@domain, for example, sales@manifold.net

Addresses in the form "Manifold Sales" <sales@manifold.net> will not work.

Skip duplicates Email only once to each email address if more than one selected record has the same email address. Checked by default.

Subject Enter text to appear as the subject field for each email.

Message Enter a text message to appear as the body of each email. If we have a longer email, it may be more convenient to compose the message as a text file and to then copy from the text file and to paste into the dialog. A Manifold project might have several email texts on hand in the form of comments components that can be copied and pasted into the Message dialog box as desired.

Expand column references Expand escape sequences containing column names to column values. Checked by default.

OK Send the emails.

Cancel Exit without sending email.

The Subject and Message texts can contain the names of columns in square brackets [ ]. Sending email messages will automatically replace the column names with data from those columns from the relevant table records.

To send email from Manifold:

1. Open a drawing or table that contains a field with an email address.
2. Select the objects in the drawing or the records in the table to which email should be sent.
3. Choose Send Email from the main Drawing or Table menu.
4. Fill in a value for the From field (will be overridden by the Outlook default From account setting if Outlook is used).
5. Choose the column name that contains email addresses for the To setting.
6. Enter whatever text is desired for the Subject and Message boxes and press OK.
7. Email message will automatically be created for each selected object and loaded into the Outbox for Outlook.

Example

Let's imagine that John's Gas Stations is a company operating a chain of gas stations in Northern California. The company wishes to send an email message to each congressional representative in whose district they operate a gas station to urge opposition to a proposed law that would force the company to raise prices.

We begin with a map that has two layers: a drawing layer called Congress showing congressional districts and containing the email address for each congressional representative, and a drawing layer called Facilities that shows the location of our gas stations.

With the focus on the Congress layer, we select the desired districts in the Congress layer using the above settings in the transform toolbar.

The result is that each congressional district containing one of our gas station facilities has been selected.
If we open the Congress drawing's table (with the selection filter button pushed in so that only selected records appear) we can see the data for the selected districts. Note that a field called Email contains the email address for each representative.

We launch the Send Email command and fill it the boxes as seen above. Note the use of field names in square brackets [ ] that will be substituted for each email with the appropriate value for that record. For example, the subject text in the email sent to congressman Pete Stark will have the Subject text of

**Important Message for Pete Stark**

and the Message body in the email will begin

**Dear Representative Stark,**

The use of square brackets and appropriate data fields provides a degree of personalization for what might otherwise be read as an impersonal mass mailing.

When we press OK the system will create six email messages, one for each selected object, and will pass them through the Windows MAPI layer for sending. If we have Outlook installed, the messages will end up in the Outlook mailbox, ready for sending.

**Emails Must be Valid**

Send email can only work with email addresses that are correct email addresses. If the table used contains an invalid email address, the send email process will stop with an error message when it hits that record.

For example, if sending email to sales@manifold.net a typographic error that inserts an unnecessary space as in **sales@manifold.net** or **sales@manifold.net** would result in an invalid email address that will stop the send email process.
An easy way to find invalid email addresses is to use the Query Toolbar with the not button pushed in and the Containing Match to search for the regular expression given below:

```
({\w.\!$%\-+}+@[A-Za-z0-9\-]+(\.[A-Za-z0-9\-]+)+)
```

Pressing Select in the query toolbar will find all records that do not have valid email addresses. Using the selection filter in the table to show only selected records, the malformed email addresses may then be corrected or their records deleted.

As an alternative, we can use the Edit - Find command with the regular expression box checked to search through email field using the regular expression given above.

Copy and paste the above regular expression from this Help topic into the Find box. The regular expression will select records that have valid email addresses. Using Edit - Select Inverse we can invert the selection to find all those records not matching the above regular expression.

**Using Outlook and Outlook Express**

When sending a lot of mail, the system works much faster if Outlook or Outlook Express has already been launched before Send Email begins sending mail messages.

The emails will be created and placed in the Outlook Outbox. They will be sent the next time Outlook sends and receives mail. If you have Outlook set up to automatically send and receive mail using a full-time Internet connection, the email will go out with the next send/receive operation. If you are using a dial-up line, connect to your ISP and check mail as you normally would and the emails will go out at that time.

Emails will be passed to Outlook and Outlook will do the actual mailing. If you look in the Outbox before the mail gets sent and see a name instead of an email address for a record, check your Outlook Contacts to see if you have that particular email address entered as a Contact with a different name in your Outlook installation. In such cases Outlook is taking the email address passed by Manifold and substituting a name based on the Contacts information. The mail will still be sent properly; Outlook is just trying to be helpful by showing you the name associated with that email address.

When Outlook is installed, the From field will be inherited from however you have set up Outlook. There are methods to alter the From field by establishing an account with a fake originator address and (temporarily) setting this account to be the default for Outlook; however, if you are planning on doing such things you already know how to do this. Please don't use such techniques for spam purposes.

Outlook and Outlook Express may raise a confirmation box when Send Email is used to confirm that the user allows sending of email from a different program. This is an anti-spam measure that is intended to prevent viruses or other malign programs from using a computer's Outlook installation to send spam without the user realizing it. However, such confirmation boxes can be highly annoying when sending lots of email using Send Email.

Mail confirmation message boxes displayed by the Outlook Express can be disabled by checking off the "Warn me when other applications try to send mail as me" option in the Outlook Express Tools - Options - Security menu. Unfortunately, Outlook XP does not appear to have the same option (a good reason to use Outlook Express when working with Send Email).

**Paging**

Keep in mind that the ability to send email is also the ability to page someone via email. If we keep the right pager email addresses in the map's data fields, we can use Send Email to page people. Cool!

It is important to know the terms and conditions the pager service provider has for their email to pager service. An email paging service may require us to know the pager's PIN number as well as one or more numeric identifiers for that pager. For example, to send a page, one might be required to use an email address in the form

```
8007208398.9801155@pagercompany.com
```

Pay attention to the maximum message lengths acceptable to the pager service provider. Some providers will take long messages and split them into multiple pages sent to the pager. In such cases, the messages we send to the pager should be kept short.

Not all email to pager services can handle all of the fields that may appear in a normal email message. Some services cannot handle a Subject: line while others encourage it.

Note that not all pager companies allow pages via email. Some companies provide this service only to national customers and do not offer it to regional pager service customers. Other companies do not provide it at all. Some companies say they have an “email to pager” service, but it is a restricted service that does not allow general email as is commonly used throughout civilized society.

Examples of annoying restrictions include required use of proprietary pager company email packages, special web mail forms on web pages or the purchase of a special email account from the vendor. Some companies say they have “email to pager” service when what they have is only a form on their Internet web page but no access from normal email. Paging services are a highly competitive market. If your pager service company does not have true email capability, dump them and sign up with a company that provides the service you want.

If you are having trouble sending a page from within Send Email, try using the same addresses and body text from the same machine using Outlook. If you cannot use Outlook to send a message to the desired pager, this indicates a problem with your computer’s Outlook configuration, with your Internet connectivity, or with the email to pager service. If Outlook cannot send an email to the pager, Send Email will not be able to either.

A Request from manifold.net

GIS has played a key role in the development of “direct mail” in snail mail. GIS will also play a key role, for better or for worse, in the development of sophisticated geo-targeting methods for email as well as snail mail. We’d like to ask our colleagues and customers to use Send Email only for beneficial uses and in situations where the recipient wants to be contacted. Please do not use Send Email to spam people.

Send Email can be used for many good purposes. Emergency coordinators can be notified if a storm threatens their community. Volunteer fire department members in rural areas can be paged to deal with a specific fire threat. Companies can send email to representatives or dealers within a certain drive time of a proposed new product tour. The possibilities for good are endless.

Targeted email for those who want to receive it is good. Spam is bad. Please, let us all make sure that we in the GIS community will not be the ones who ruin a good thing.

Notes

The John’s Gas Stations example is fictional. Perhaps the most fictional aspect of the example is the use of email addresses for congressional representatives. Very few of the representatives have published emails for the public to use. Most hide behind a web form and do not allow direct emails.
**Topology Factory**

The **Drawing - Topology Factory** menu selection is enabled when the optional Business Tools extension has been installed and the focus is on a drawing window. If you have not activated the **Business Tools** extension with a valid serial number you will not be able to use the **Drawing - Topology Factory** command.

The **Drawing - Topology Factory** command provides an interactive dialog to view and repair common topological errors in drawings. To use Topology Factory, open a drawing and then launch Topology Factory. The system will scan the drawing and report any incidents found in a list, grouping incidents of the same type together. If a selection is present, Topology Factory works only on the selected objects.

Selecting a particular type of incident, such as **Lines intersecting other lines** will choose that set of incidents for repair. If there is more than one such incident of that type, the system will choose the first incident of that type and will make one of the lines involved in the incident red and the other line blue.

Clicking **Center incident** will zoom into the current incident. Clicking **Previous**, **Next** or **Last Incident** will choose one of the other incidents of that type.

Clicking **Fix Incident** will repair the incident using the method specified, where **auto** tells Manifold to choose which line to edit, and choosing **modify blue** or **modify red** will specify which line is to be modified so that the two lines do not intersect each other.

Topology Factory will report many relationships between objects as incidents that may or may not be errors depending on the intent of the drawing. For example, if we intend that some lines will intersect areas or that some areas are to overlap other areas then such incidents are not errors. On the other hand, if we do not want any areas to overlap other areas then such incidents are errors to be repaired. Since Topology Factory does not know what we intend it reports all incidents within the repertoire of incidents for which it scans.

To make it easy to see detected incidents out of a list of many possible types of incidents, Manifold provides four buttons to control sort order that may be used to rapidly bring relevant incidents to the top of the incidents pane.

**Location Precision**

The **Location precision** parameter in the **View - Properties - Precision** dialog guides Topology Factory in deciding when an object is sufficiently "near" another object to trigger an incident. The **Location precision** factor should be set to a distance value so that proximity of objects over distances smaller than the specified distance will be detected as an incident. For example, if desired, the **Location precision** factor may be relaxed to a higher value while working with Topology Factory to detect and correct incidents and then set back to a smaller value for more detailed work. For convenience, Topology Factory provides a **Precision** box to specify precision within Topology Factory.

Note that lowering the value of the location precision parameter (making it a smaller number) will increase the precision, that is, it will cause Topology Factory to examine smaller details. Increasing the value of the location precision parameter (making it a larger number) will cause Topology Factory to examine incidents over larger details. Setting the location precision parameter to a larger value, for example, will cause Topology Factory to ignore incidents where lines overshoot areas by very small distances that are less than the large size of the location precision parameter.

As with the case when working with any automated tool, keep in mind that making sweeping changes with Topology Factory using a precision parameter that is significantly larger than the precision with which the drawing was originally created may make undesirable changes in the drawing. For example, lines that intersect at a very acute angle may have their point of intersection moved. To avoid unintended changes in such cases, take a moment to save the project before making changes. Examine incidents carefully and make changes only after viewing each incident. Although Topology Factory is quite good at identifying incidents based on the precision parameter in use, it cannot read our minds to know in all cases what we intend. That is why an interactive dialog is provided to make it easy for the user to review incidents and to decide what should be done in each case.

**Controls**

(Visual preview pane) Displays the drawing with the current incident marked in red and/or blue selection color.
Zoom To Fit - Zoom so that the entire drawing fits within the preview pane.

Zoom In - Magnify the view as if seen from a closer distance.

Zoom Out - Reduce the view as if seen from farther away.

Zoom Box - Zoom to the size of the cursor box drawn with the mouse.

Center Point - Pan the view so that the spot clicked is centered.

Grabber - Interactive pan: click and drag with the grabber hand. The scene will be panned so that the initial point is moved to the spot where the drag is released.

Center Incident - Pan and zoom the display to the current incident.

First Incident - Choose the first incident for repair of a set of incidents.

Previous Incident - Choose the previous incident for repair of a set of incidents.

Next Incident - Choose the next incident for repair of a set of incidents.

Last Incident - Choose the last incident for repair of a set of incidents.

(fix method box) Choose a method to repair this particular incident: auto - Manifold chooses. modify blue - modify the object shown in blue. modify red - modify the object shown in red.

Fix the current incident using the selected method.

Fix all incidents of the highlighted type using the default fix method (auto).

(Incidents list pane) A list of incidents found in the drawing, with like incidents grouped together in the same type of incidents. Incident types with more than one incident will be shown in boldface together with the number of such incidents. If the repair of a given incident may affect other incidents, the number of incidents for such possibly affected types will be shown with an asterisk to indicate that they should be refreshed.

Alphabetic - Sort the incidents list pane by alphabetic order.

Alphabetic, Blanks Last - Sort the incidents list pane by alphabetic order, grouping detected incidents at the top of the list.

Recommended - Sort the incidents list pane by items Manifold recommends we should inspect.

Recommended, Blanks Last - Sort the incidents list pane by items Manifold recommends we should inspect, grouping detected incidents at the top of the list.

Refresh Re-scan the drawing for possible incidents that need to be repaired.
Types of Incidents

Topology Factory identifies the following:

**Areas near other areas**  Areas that are within the precision parameter of other areas, including overlaps.

**Areas near themselves**  Areas where any portion comes near itself within the precision parameter. Often will flag multi-branched areas with islands or holes when these disjoint portions of the same area are "near" each other.

**Areas overlapping other areas**  Areas that overlap other areas where the overlap is at least the precision parameter.

**Areas overlapping themselves**  Areas that overlap themselves (possible to do with pathological topology) where the overlap is at least the precision parameter.

**Areas with multiple islands**  Areas with multiple branches that form disconnected islands. Holes in multi-branched areas are not islands.

**Areas with redundant or invalid metric**  Areas where the sequence of coordinates defining the area is either redundant (for example, repeated identical coords) or forms an invalid metric. Multi-branched areas are flagged as possible invalid metric, so areas with islands or holes will be reported.

**Lines intersecting areas**  Lines that cross over areas, having an intersection of at least the precision parameter.

**Lines intersecting other lines**  Lines that cross over other lines.

**Lines intersecting themselves**  Lines that cross over themselves.

**Lines near areas**  Lines within the precision parameter of an area.

**Lines near other lines**  Lines within the precision parameter of another line.

**Lines near themselves**  Lines for which any portion comes near itself within the precision parameter. Often will flag multi-branched lines when these disjoint portions of the same line are "near" each other.

**Lines overlapping other lines**  Lines that coincide with another line.

**Lines overshooting areas**  Lines where the end of the line extends past the boundary of an intersected area.

**Lines overshooting other lines**  Lines where the end of the line extends past an intersection with another line.

**Lines overshooting themselves**  Lines where the end of the line extends past an intersection with a different portion of the same line.

**Lines undershooting**  Lines where the end of the line does not quite reach the boundary.
areas of an area.

Lines undershooting other lines
Lines where the end of the line does not quite reach another line.

Lines undershooting themselves
Lines where the end of the line does not quite reach a different portion of the same line.

Lines with contiguous branches
Multi-branched lines where some branches coincide.

Lines with multiple separate branches
Multi-branched lines where branches do not coincide.

Lines with redundant metric
Lines composed of sequences of coordinates where some coords are repeated or are otherwise unnecessary.

Points coinciding with other points
Points that are located at exactly the same position as another point.

Points near areas
Points that are located within the precision parameter of an area.

Points near ends of lines
Points that are located within the precision parameter of the end of a line.

Points near lines
Points that are located within the precision parameter of any part of a line.

Points near other points
Points that are located within the precision parameter of another point.

Incident List Sorting

The incident list may be sorted using the four buttons immediately above it as follows:

Pressing the Alphabetic button sorts incidents by alphabetic order.

Pressing the Alphabetic, Blanks Last button sorts incidents by alphabetic order, bringing found incidents to the top of the list.
Pressing the **Recommended** button sorts incidents by priority in which Manifold recommends they be fixed.

**Example**

Consider a drawing with two areas and two lines. The two areas are colored in different colors using thematic formatting.

The areas have been drawn so they overlap.
When we launch **Drawing - Topology Factory** the dialog opens showing the first highlighted incident centered at higher zoom so we can easily see the incident. When two objects are involved in an incident Topology Factory will draw one of them in red and one of them in blue so that we can easily choose which object we would like to modify to resolve the incident.

We have configured the incidents list using the **Alphabetic, Blanks Last** sort option so that found incidents are displayed at the top of the list. We have highlighted the **Areas overlapping other areas** incident.

In the upper right corner of the dialog we choose **modify blue** and then click the **Fix Incident** button.
Manifold will redraw the display, showing the object that has been modified in green. We can click **Refresh** to recompute topological relationships (which might have changed as a result of the modification).

After the refresh, we can see that the blue area has had the overlapping portion removed.

If we take a look at the incidents list, we see that there is no longer a report of overlapping areas. Instead, the first item on the list is **Areas near other areas**, which makes sense because the two areas are now adjacent.
In the above way we can proceed through the list of all incidents reported by Topology Factory, examining each and deciding how we would like each reported incident to be resolved.
Images

Images in Manifold are usually the familiar sorts of image that almost everyone has viewed or edited with applications such as Microsoft Paint, Microsoft Image Editor, Adobe PhotoShop, Microsoft Photo Editor and similar programs.

Such images are commonly stored as TIF, BMP, GIF, JPG or various other common image formats. Existing images may be imported into a Manifold project using the File - Import - Image command. New images may be created in Manifold and then exported into common image file formats using File - Export.

Creating a New Image

1. Open a project or create a new project with File - New.
2. Choose File - Create - Image to launch the Create Image dialog.
3. Provide a name for the image, the desired size in pixels and the type of image and press OK.

The File - Create - Image dialog takes the projection parameters for the new image from whatever window is active at the time the new image is created. If the project pane is active, the image will be created using the system default projection of Orthographic centered at the 0,0 world latitude/longitude origin. If a map window or other image window is active when the new image is created, the new image will be created using whatever projection is used by that active window. This context-sensitive setting of default projection parameters makes it much easier to create new images using projection parameters that are hassle-free by default.

Linked Images

An image that is imported into Manifold is stored within the Manifold .map project file. An alternative to importing images into Manifold is to link images into the project using the File - Link - Image command. Images may be linked into a Manifold project from a variety of sources outside of the .map file. When images are linked into a Manifold project the .map project file does not contain the actual image but rather contains a pointer to the source of the image, which remains stored outside of the project file.

See the Linked Images topic for additional information.

Image Fundamentals

Images are made up of pixels ("picture elements") arranged from left to right in a series of equal rows. They are often referred to as raster files in older GIS applications. ("Raster" is the old television word for the sweep of an electron gun across a series of rows used to create an image.)

Although images are normally photographic or scanned images that represent visual images of the world around us, images in Manifold can also be abstract data sets that are saved as pixels in a raster arrangement. For example, a multi-spectral satellite sensor might create an "image" of a continent from space where each pixel contains numbers representing temperature, reflectance, height above sea level as well as other information. Pixels in such an image may be colored according to their data values to convey the data content of the raster data set in a visual way.

Let's begin by explaining images in the context of familiar sorts of photographic images.
Images are fundamentally different from drawings in several ways:

- Images are made up of a sea of pixels and contain no "objects." Any objects we perceive in an image are the sole result of our psychological interpretation of regions of pixels of like color to be distinct things. Zoomed up close the image is a blur of pixels. Zoom far enough into the image above and it is unclear where the "sky" ends and the "monument" begins. In contrast, drawings contain true objects that are defined by precise sets of geometric coordinates for each object.

- The appearance of an image is highly dependent upon the zoom level. At zooms greater than 100%, the individual pixels will become visible as the image becomes ever more like a coarse mosaic. Drawings, on the other hand, are always razor sharp no matter how high the zoom. Objects in drawings will not change their appearance when zooming in or out: if a point is formatted as a small dot the dot will appear to be the same no matter what the zoom. In a drawing an intersection between lines will always be razor sharp.
whereas in an image what might appear to be a sharp intersection will at high zoom be seen as clumps of pixels.

- Images have no intrinsic coordinate system embedded in the image but rather depend upon an implied coordinate system suggested by the relationship of pixels to each other that is defined by their arrangement in rows. Drawings have an implicit coordinate system implied by the coordinates that define the objects. If an image is to be used within a coordinate system, it must first be referenced to that coordinate system.

- When drawings are re-projected the number of objects stays the same even though their shape may be different. When images are re-projected pixels are always created or deleted through interpolation as the images is projected into a different shape.

**Compressed Images**

Compressed images are a special type of image that is dynamically reconstituted as needed from compressed data. Although they may appear to consist of pixels and may be described as consisting as a certain number of pixels in width and height, they are dynamically created as necessary and do not actually consist of a fixed number of pixels. Compressed images are used for very large images where display must be fast and storage requirements reduced to accommodate the very large size of the image. In exchange for fast speed and reduced storage, compressed images are limited to display only functions and may not be edited.

**Georegistering Images**

Georegistering is the process of telling Manifold where a particular image is to be located on Earth. See the Coordinates Tutorial for more information on the concepts behind this process. Georegistration is closely related to the idea of projecting an image.

By default, images in Manifold are taken to be in Orthographic projection located at zero degrees latitude and zero degrees longitude, with each pixel in the image assigned a true geographic location based on the given size of pixels or DPI used when the image was created or imported. It does no harm to assume a geographic location for images that will be used in a purely abstract way (like PhotoShop), and having a built-in assumption about projection and location is very useful when images will be used in a geographic context.

See the Georegistration topic for information on how to georegister an image for use within a map. See the Projections and Images topic for more information on use of projections with images.

**Different Types of Images**

Because images can involve very large numbers of pixels and each pixel can involve several numbers to describe its color, images can require very large amounts of storage space. Many methods have been invented to save space when storing images. The more popular methods have become so prevalent that most image processing systems, including Manifold, allow working with images that are internally structured using several different standard methods. These appear in the system as several standard types:

- **Grayscale Images** - These are coded using one number per pixel representing one of 256 different gray tones ranging from black to white.

- **Palette Images** - These are images coded using one number per pixel, where the number specifies which color in a palette of up to 256 different colors should be displayed for that pixel. The colors in the palette can be True Color RGB colors. Palette images save space at the cost of a reduced total number of colors available for use in the image.

- **RGB Images** - These images use three numbers for each pixel, allowing possible use of millions of colors within the image at the cost of requiring three times as much space as grayscale or palette images. They are often called True Color RGB images in Microsoft applications.
• **RGBa Images** - These images are RGB images with a fourth number added for each pixel that specifies the transparency of that pixel in the range 0 to 255. RGBa images are used when combining multiple images in maps for elaborate graphics composition or creation of special visual effects in maps.

• **Compressed Images** - Compressed images use sophisticated wavelet compression technology to not only compress the amount of data an image requires but also to reconstitute the image on the fly on demand. At any given zoom level the desired view of the image is reconstituted from the compressed data store. Compressed images in general may be viewed but not altered or otherwise manipulated.

In addition Manifold can import data from multi-spectral raster data images that contain many channels. When importing from formats that support many channels Manifold will import each channel into one of the above image types (most normally, as a grayscale image). Operations on multi-spectral raster data sets can then proceed by choosing those images/channels to use. For example, three of the imported images can be combined into a single RGB image to create a "false color" image that uses three images/channels as R, G and B channels in a single image.

For many operations the various types of images will seem equivalent, except that compressed images cannot be edited or manipulated. We can select a rectangular region of pixels in an non-compressed image using Select Box, for example, without it mattering what type of image is involved. Some operations will only work for certain types of images. For example, the Hue / Saturation command only works with RGB or RGBa images and so this command will be disabled whenever the focus is on a grayscale image.

We can always convert a grayscale or palette image to RGB in order to use a particular command and then convert it back. To convert an image from one type to another, use the Image - Convert To dialog.

**Transparency in Images**

Drawings consist of empty space in any region that is not occupied with a point, line or area. In contrast, images have no empty space. Every part of an image is filled with pixels. To allow "see through" regions in images, Manifold images may have invisible pixels through which any items in lower layers may be seen. Invisible pixels are simply placeholders that do not appear. It is if there are no pixels in that part of the image. Invisible pixels are often used to give the appearance that images are not rectangular but consist of some irregular, non-rectangular shape. However, every image is rectangular because every image consists of a series of equal-width rows of pixels.

To make part of an image transparent, we select the desired region and then Delete those pixels. Invisible pixels may be used with any of the four types of images.

Any image layer can be made partially transparent by changing its opacity from 0% to 100% in steps of 1 percent using the controls in the Layers pane. This is a great way to create spectacular effects. Transparent layers work with all types of images.

In addition, RGBa images can have each individual pixel assigned a percent pixel transparency value. This is normally accomplished through partial erasers and other tools. When combined with Manifold image editing tools this effect can be used to compose amazing images by combining many layers of other images. See the Layers topic for an example.

To summarize, there are four types of transparency that may be found in a map:

- Drawings are transparent in empty space not occupied by an area, line or point.
- Any image may be made fully transparent in regions of invisible pixels.
- Each pixel in an RGBa image may be partially transparent using RGBa pixel transparency.
- Any image layer may be made partially transparency using layer transparency.

**Image Windows vs. Map Windows**

An image may always be popped open in an image window to show the image by itself using the native projection used for that image. This is a handy way to see the image in its native state if there is any confusion about how it appears in a map.

Map windows are the normal user interface for working with images mainly because of the layer capabilities of maps. Maps can include many image layers and thus allow the use of many simultaneous images. In graphics...
composition we will often work with many image layers at the same time to compose an image by stacking elements.

The illustration above shows four image layers above several drawing layers in a map. The Manifold logo and text are both in separate layers and their shadows underneath are drop shadows in RGBA image layers created using Gaussian Blur. Both drop shadow layers have had opacity decreased so their shadows are not so obvious.

**Using the Layers Pane with Images**

The Layers pane is used to control the appearance of images within image windows. The layers pane includes checkboxes for two system "layers" - a background color layer and a border layer that shows an enclosing box about the height and width of the image.

By default, images are shown using the checkerboard background Manifold uses to provide a backdrop for any transparent regions. The layers pane is shown to the right of the image window. Illustrated is the standard **bronze** image with regions other than the monument selected and then deleted into Invisible Pixels.
Checking the **Border** box will draw a one-pixel border around the height and width of the image. This is a good way to see the actual extent of an image that contains regions of invisible pixels.

Checking the **Background** box in the layers pane will replace the checkerboard background with whatever is the default background color. This is a good way to see the actual extent of an image that contains regions of pixels that are the same color as the background (usually white).

Note that only maps can have true "layers" in Manifold in the sense that they can layer more than one component within the same map window. The border and background "layers" in the Layers pane for images are not true layers even though they appear in the Layers pane in the same manner as do layers in maps. These are simply system controls that take advantage of the Layers pane as a conceptually convenient user interface.

Checkboxes show above other than **Border** and **Background** are explained in the Images and Channels topic.

**Layouts and the Layers Pane**

If an image has any Layouts created they will appear as "layers" in the layers pane for that image. Checking the box for one of these print layout layers will cause a layout rectangle to appear in the image that shows the region covered by the layout.

Right clicking onto the hatched border of one of the layout rectangles in the image will cause a context menu to appear with controls based on that layout rectangle. For example, we can **Zoom** to a given layout rectangle, **Print** it or change its **Properties**. If a layout is empty (for example, if the layout scope is set to **selection** and no pixels are selected in the parent image) zooming to the layout will do nothing.
Use **Tools - Options - Colors - Layout Rectangle** to change the color in which layout rectangles are shown. The default color is black.

**Nomenclature**

Transparency and opacity are two terms that mean the same concept viewed from different directions. When something is completely opaque it is not at all transparent. When something is perfectly transparent it may be said to have zero percent opacity.

Which word is used depends on the discussion. When imagining layers stacked up above each other like transparent sheets it is conceptually clearer to use the word **transparency**. When discussing a specific percentage of light transmission to be applied via a slider bar in a dialog most applications use the word **opacity**.

The convention in the graphics arts editing software industry is to adjust layer transparency with controls that set a number from 0% to 100% opacity, so that an image with 100% will be fully opaque and not allow any view of an image underneath it. **Manifold** follows this convention. This convention persists in the graphics arts industry even though the technical implementation of transparency effects is done using an **alpha** channel within RGBA images where the higher the value of the alpha channel (from 0 to 255) the higher the transparency.

One therefore encounters the slight conceptual dissonance of increasing opacity with higher numbers (up to 100%) in dialogs and other user interfaces while the internal data sets use numbers (alpha channel values) in which opacity decreases with higher numbers. Since we rarely set alpha values by hand this is not so bad. Alpha values are normally set using various tools, such as erasers, or masks. In the case of masks, the darker the mask region the lower the alpha value is and thus the higher the opacity. From a casual conceptual view this is very acceptable because it leads to an effect where black regions of masks cause full opacity and white regions of masks cause full transparency. Since we are used to thinking of "white space" as being transparent this works well as a natural mnemonic for the effects of masks.

**Images can be Huge**

Images are often saved on disk in compressed file formats such as .tif or .jpg or even within a project saved in **Manifold’s compressed .map** format. However, once the image is loaded into Manifold it must be uncompressed so that every pixel is available for viewing, selection and possible manipulation or alteration as a result of editing or georegistration or other processes. A 50MB, compressed .tif file can easily be expanded into a 150MB uncompressed image within Manifold when it is uncompressed from the storage format.

One way to get an idea of the active size of images is to consider their size: RGB images require 4 bytes for each pixel. A 6000 x 6000 pixel RGB image therefore requires 144 megabytes. When working with four such images at once we will have almost 600 megabytes of image space even before any temporary files are considered.

One strategy to manage very large images is to use compressed images, a special type of image that always remains compressed and is dynamically reconstituted as necessary. Compressed images require much less space either in the **Manifold** project or when saved on disk, and they allow very fast redisplay. However, the cost of using **compressed images** is that they can only be displayed - they cannot be edited or otherwise manipulated.

Unless the images are compressed images, to work with large images you **must** have large amounts of RAM and large amounts of free space on disk. You should also use a reliable operating system such as Windows XP or Windows 2000 that can adequately handle large amounts of RAM. Older Windows editions such as Windows ’98 have so many memory-management bugs that they cannot reliably be used for tasks that require large amounts of RAM. Since RAM is very inexpensive, load your computer with enough RAM so that the entire project can fit in RAM memory. Ideally, load a full two gigabytes or more.

For **Manifold** to work your machine must have adequate RAM, an operating system that can handle the necessary RAM, or adequate disk space. If any of these factors are not present your system will report various error messages when working with images or other large tasks.

**Other Image Topics**

Projections and Imported Components - Discussion of projections issues and formats when importing images plus simple explanation of channels.

**Palettes** - How palette images work.
RGBa Pixel Transparency - RGBA explained, with examples.
Projections and Images - More advanced discussion of projection issues with images. Read after reading the Projections and subsequent projection and coordinates topics.
Compressed Images - Compressed images are used for display of very large images.
Linked Images - Linked images are dynamically created from queries or tables.
About the Sample Images provides notes on the bronze monument and other sample images.
Image Window Menus and Controls

This topic lists default menus and controls that appear when an image window is active. These will also appear when an image tab is active in a map. Image commands are all listed under the Image menu. Some commands are organized under submenus called Adjust and Effects. These somewhat arbitrary categories are used so Manifold users who are familiar with Adobe PhotoShop will find the menu structure more familiar when seeking analogous commands.

This list also calls out the most popular menu items, such as from the View - Panes menu, that are especially relevant to use with images.

Many image editing commands work only with RGB or RGBa images. See the Image Types topic. Image editing commands are not available for compressed images. To edit a compressed image, first convert it to an RGB or RGBa image, edit it and then convert it back to a compressed image.

Edit Menu

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cut</td>
<td>Windows clipboard cut operation. Copy pixels to the windows clipboard and delete them from the image.</td>
</tr>
<tr>
<td>Copy</td>
<td>Windows clipboard copy operation. Use this to copy pixels from the image onto the clipboard for pasting elsewhere.</td>
</tr>
<tr>
<td>Paste</td>
<td>Paste the contents of the Windows clipboard into the image and deletes any selected pixels. Creates new pixels in the image.</td>
</tr>
<tr>
<td>Paste Append</td>
<td>Paste the contents of the Windows clipboard into the image without deleting any selected pixels. Creates new pixels in the image.</td>
</tr>
<tr>
<td>Delete</td>
<td>Permanently delete selected pixels from the image.</td>
</tr>
<tr>
<td>Delete All</td>
<td>Permanently delete all pixels from the image.</td>
</tr>
<tr>
<td>Assign Projection</td>
<td>Used with drawing, image, labels and surface windows to change the interpretation of the data. This is a specialized function that is applied only when manually specifying the projection of a component imported from a format that does not correctly store projection information. To change the native projection of any component other than a map, use the Edit - Change Projection command.</td>
</tr>
<tr>
<td>Change Projection</td>
<td>Used with components like drawings and images to re-project the component and thus permanently change the data.</td>
</tr>
<tr>
<td>Select All</td>
<td>Select all pixels in the image.</td>
</tr>
<tr>
<td>Select None</td>
<td>Deselect all pixels in the image.</td>
</tr>
<tr>
<td>Select Inverse</td>
<td>Invert selection: those pixels that were selected will be deselected while those pixels that were not selected will be selected.</td>
</tr>
<tr>
<td>Select Mode</td>
<td>Choose the selection mode to be used for mouse selection:</td>
</tr>
<tr>
<td></td>
<td>Replace - Any selection made with the mouse will replace the previous selection.</td>
</tr>
<tr>
<td></td>
<td>Add - Any selection made with the mouse will be added to the previous selection.</td>
</tr>
<tr>
<td></td>
<td>Subtract - Any selection made with the mouse will be subtracted from the previous selection.</td>
</tr>
<tr>
<td></td>
<td>Invert - Any selection made with the mouse will be inverted with the previous selection.</td>
</tr>
<tr>
<td></td>
<td>Intersect - Any selection made with the mouse will be</td>
</tr>
</tbody>
</table>
intersected with the previous selection.

**Modify Selection** Changes the shape of regions of selected pixels.
- **Border** - Select those pixels on the border of the selected region(s).
- **Contract** - Uniformly contract selected region(s) by given number of pixels.
- **Expand** - Uniformly expand selected region(s) by given number of pixels.
- **Smooth** - Provide a smoother edge to the region(s) of selected pixels.

**Snap To** Enable / Disable snap modes:
- **Graticule** - Snap to graticule intersections if a graticule is displayed.
- **Grid** - Snap to grid intersections if a grid is displayed.

**Save Mask/Channel** Create masks from the current image or save channels as grayscale images.

**Load Mask/Channel** Load a mask or a channel from a grayscale image.

**Go To** Go to the entire image or to a selection.

**View Menu**
- **Back** Go back one view.
- **Forward** Go forward one view.
- **Zoom In / Out** Zoom in brings us closer to the image while zoom out moves us farther away.
- **Zoom To Fit** Zoom so that the image fits the existing window size.
- **Zoom To** Zoom to a specified zoom level.
- **Graticule** Show a graticule (latitude / longitude lines).
- **Grid** Show a grid that may be used to measure or to align editing and snap commands.
- **Legend** Display a legend showing formats.
- **North Arrow** Display a North arrow.
- **Scale Bar** Display a scale bar.
- **Display Options** Enabled for compressed images. Enables choice of which channels will be used as R, G, B and alpha channels.
- **Panes** Call a dialog that allows displaying or hiding all panes.
- **Full Screen** Show the image over the entire monitor area.
- **Refresh Data** Update a linked image with the latest data from the controlling geocoded table.
- **Refresh View** Update the visual appearance of the image with any changes made since the last refresh. Enabled only if **Autorefresh View** is off and some change (such as a selection) has been made that might affect the visual appearance of the image.
- **Autorefresh View** Automatically update the image whenever any changes are made that might affect the visual appearance of the image. Turn this option off when working with very large
images where there is no desire to take the time to show, say, a selection when it is made (repainting a very large image on every selection step might take longer than desired). When **Autorefresh View** is off, we can always cause a refresh on demand using the **Refresh View** command.

**Properties** View the properties dialog for this image.

**View - Panes Menu** Panes of special interest when working with images include:

**Layers** When used with image windows, the layers pane turns individual R, G, B or alpha channels on and off and also controls whether the image border or background is displayed. See also the Images and Channels topic.

**Selection** Save and recall regions of selected pixels.

**Tool Properties** Specify characteristics of image editing tools, such as the opacity of a paintbrush.

**Image Menu**

**Open Palette** Enabled for palette images. Open the palette for this

**Open Data Source** Open Database Console focused on the data source of a linked image.

**Dither** Uses a smaller number of colors in an image while preserving a perceptual effect through the use of dither patterns.

**Quantize** Convert image to the specified number of colors using enhanced or standard methods.

**Resize** Change the size of the image using desired interpolation method.

**Convert To** Convert images between grayscale, palette, RGB, RGBA or compressed images.

**Image - Adjust Menu**

**Color Balance** Adjust RGB color values for Shadows, Midtones or Highlights

**Brightness / Contrast** Change brightness and contrast.

**Hue / Saturation** Alters hue, saturation and lightness.

**Invert** Invert pixel channel value. Creates a "photographic negative".

**Equalize** Alters individual pixel brightness so there is an equal number of pixels at all brightness levels.

**Gamma** Change mid-tone color brightness.

**Threshold** Forces grayscale pixels to white, black or unchanged based on a histogram.

**Threshold Color** Like **Threshold** but applies to each RGB channel individually.

**Posterize** Convert image to a limited number of colors for a "poster"
Image - Effects Menu

- **Colorize**: Adjust saturation and hue while keeping intensity the same, as if a color were applied to a grayscale version of the image.
- **Gaussian Blur**: Apply Gaussian bell curve blurring to create shadows, halos and transitions.
- **Motion Blur**: Streaked effect in the desired direction.
- **Diffuse**: Moves pixels by swapping them in a "random walk."
- **Filter**: Applies convolution filters, either custom filters or a choice from numerous presets.
- **Fluoresce**: Alter saturation and lightness in asymmetric ways to cause some colors to appear unusually bright.
- **Noise**: Add random monochromatic or color pixels.
- **Relief**: Add highlights and shadows to give a 3D effect to 2D images.
- **Simplify**: Aggregates pixels into larger clumps for a pointillist or mosaic tile effect. Can also remove small dots in cartographic work.
- **Tile**: Convert image into rectangular tiled regions of
- **Relink**: Re-links a linked image to a data source.
- **Unlink**: Converts a linked image to a local image with no connection to the originating data source. Linking and then unlinking an image is equivalent to importing it.
- **Create Index Drawing**: Creates an index drawing for an image library.
- **Download**: Downloads intermediate levels of an image linked from a Manifold image server, OGC WMS server or TerraServer.

Transform Toolbar

- **Operators**
  - **Add Margin**: Add to the rectangular size of the image by the given number of pixels at the margin.
  - **Add Noise**: Add color noise as set by the source / argument box.
  - **Add Noise (Mono)**: Add monochrome noise as set by the source / argument box.
  - **Auto Contrast**: Automatically adjust contrast to balance contrast non-linearly throughout the entire image considering the overall intensity of each pixel.
  - **Auto Level**: Balance contrast throughout the entire image by balancing contrast within each channel individually.
  - **Blur**: Blur image by fixed amount. Blurs both features and colors. For a "blur" effect that does not mix colors, use the Median filters.
  - **Blur (parameter)**: Blur image by the number of pixels given in the source / argument box. This blur is implemented as a matrix operator where the value of the source / argument box is the weight of the center pixel with weights of all other pixels set to 1. Increasing the value therefore produces a lesser blur.
Brightness  Adjust brightness by source / argument box amount.
Contrast  Adjust contrast by source / argument box amount.
Crop  Crop image size to the selection.
Crop Margin  Crop the rectangular size of the image by the given number of pixels at the margin.
Desaturate  Reduce saturation to zero for all color values. Remains RGB.
Difference East  Detect and emphasize edge transitions to the East. See the Filters topic for general discussion of convolution matrix filters like this one.
Difference North  Detect and emphasize edge transitions to the North.
Difference North-East  Detect and emphasize edge transitions to the North-East.
Difference North-West  Detect and emphasize edge transitions to the North-West.
Difference South  Detect and emphasize edge transitions to the South.
Difference South-East  Detect and emphasize edge transitions to the South-East.
Difference South-West  Detect and emphasize edge transitions to the South-West.
Difference West  Detect and emphasize edge transitions to the West.
Diffuse  Migrate pixels by swapping them in a "random walk." Reapplying Diffuse several times will move pixels further and further away from their original positions.
Equalize  Equalize image to value given in source / argument box.
Flip Horizontal  Flip image left / right to mirror image.
Flip Vertical  Flip image top / bottom to upside down image.
Gamma  Alter gamma value by source / argument box.
Gaussian Blur  True Gaussian probability function blur by given amount.
Grayscale  Cast all colors into gray tones by intensity. Remains RGB.
High Pass 1  Emphasize rapid transitions between pixels. Good edge detection with horizontal and vertical lines.
High Pass 2  Stronger emphasis on rapid transitions between pixels with strong edge detection on lines of all angles.
High Pass 3  Edge detection with crisp vertical and horizontal and fuzzy lines at other angles.
Invert  Invert pixel values so that 0 becomes 255 and 255 becomes 0. See the Invert topic for information on Invert At.
Laplace 1  Good edge detection with emphasis on horizontal and vertical changes.
Laplace 2  Strongest changes at point of intersection of horizontal and vertical lines, resulting at a small \( X \) pixel pattern at the point of intersection. Fuzzy...
detection of linear features at other angles.

**Low Pass 1** Strong blur, with strong suppression of large pixel to pixel changes.

**Low Pass 2** Blur, with medium suppression of large pixel to pixel changes.

**Low Pass 3** Blur, with least suppression of large pixel to pixel changes. Removes graininess in some images.

**Median Cross** Evens out color tending to preserve horizontal and vertical features. A "blur" that ignores perfectly horizontal and vertical linear features.

**Median Square** Evens out color looking at a 3 x 3 matrix surrounding each pixel. A "blur" effect that preserves prevailing color.

**Median Square 5** Strong evening out of color looking at a 5 x 5 matrix surrounding each pixel. Results in removal of isolated pixels of color differing greatly from surrounding pixels. A stronger "blur" effect that preserves prevailing color.

**Motion Blur** Four options in diagonal, horizontal and vertical directions to blur image as if moved in that direction by given amount.

**Posterize** Converts the image to a limited number of colors for a "poster" effect. Very low (2, 3 or 4) parameter values give the most obvious effect.

**Rotate** Rotate image given number of degrees. Negative degrees are taken as counter-clockwise rotation.

**Sharpen** Emphasize transitions to give images a sharper appearance.

**Sharpen (parameter)** A Sharpen with selectable value for degree of sharpness. This sharpen is implemented as a matrix operator where the value of the source / argument box is the weight of the center pixel with weights of all other pixels set to 1. Increasing the value therefore produces less sharpening.

**Sharpen More** Enhanced sharpness compared to standard Sharpen.

**Threshold** Converts an image into black and white based on whether each pixel is above or below the given value in intensity.

**Threshold Black** Like **Threshold**, but forces all pixels below the intensity value into black while leaving all pixels above that value unchanged.

**Threshold White** Like **Threshold**, but forces all pixels above the intensity value into white while leaving all pixels below that value unchanged.

**Tile** Divides image into tiles of given X and Y extent and then averages pixel values within the tiles. This is used for aggregating and interpolating raster data images into larger tiles and for creating "pixelated" artistic effects.

**Tile Median** Equivalent to **Tile** command with **Preserve Colors** checkbox checked.
**Image Types**

Manifold works with several standard types of images:

**Grayscale Images** - These are coded using one number per pixel representing one of 256 different gray tones ranging from black to white.

**Palette Images** - These are images coded using one number per pixel, where the number specifies which color in a palette of up to 256 different colors should be displayed for that pixel. The colors in the palette can be True Color RGB colors. Palette images save space at the cost of a reduced total number of colors available for use in the image. The image shown about uses only 16 colors.
**RGB Images** - These images use three numbers for each pixel, allowing possible use of millions of colors within the image at the cost of requiring three times as much space as grayscale or palette images. They are often called **True Color** RGB images in Microsoft applications.

**RGBa Images** - These images are RGB images with a fourth number added for each pixel that specifies the transparency of that pixel in the range 0 to 255. When seen in an image window, grayscale, palette and RGB images will be shown on a background of solid color (white by default). RGBa images are shown on a background of alternating white and light gray checkerboard pattern so that differences in transparency are more visible. (Use the Layers pane with images to click ON a background color layer or to show a border around an image.)

RGBa images are used when combining multiple images in maps for elaborate graphics composition or creation of special visual effects in maps. For example, the RGBa image illustrated above is shown in a layer above a grid of lines that become visible to an increasing degree as transparency increases towards the bottom of the image.
Compressed Images - Compressed images use sophisticated wavelet compression technology to not only compress the amount of data an image requires but also to reconstitute the image dynamically on demand. At any given zoom level the desired view of the image is reconstituted from the compressed data store. Compressed images can be viewed, but not edited or otherwise manipulated. Compressed images are used to display very large images that would require too much time for display and possibly too much room for storage if they were not compressed.

Compressed images have many of the display characteristics of RGBa images. The Layers pane will show Red, Blue and Green channels, and the View - Display Options command will allow choice of what image channels (in the case of multi-spectral compressed images) will be used for Red, Green, Blue and alpha channels.

Manipulating Images

Not all operations will work with all types of images. For example, compressed images may be viewed but cannot be edited, as can the other image types. For many operations the non-compressed types of images will seem equivalent. We can select a rectangular region of pixels in an image using Select Box, for example, without it mattering what type of image is involved. Some operations will only work for certain types of images. For example, the Hue / Saturation command only works with RGB or RGBa images and so this command will be disabled whenever the focus is on a grayscale image.

We can always convert a grayscale or palette image to RGB in order to use a particular command and then convert it back. To convert an image from one type to another, use the Image - Convert To dialog.

Other Types of Images

In addition Manifold can import data from multispectral raster data images that contain many channels. When importing from formats that support many channels Manifold will import each channel into one of the above image types (most normally, as a grayscale image). Operations on multispectral raster data sets can then proceed by choosing those images/channels to use. For example, three of the imported images can be combined into a single RGB image to create a "false color" image that uses three images/channels as R, G and B channels in a single image.

Saving Drawings, Maps, Surfaces and Labels as Images

Manifold can create images from other components. See the Tools - Make Image topic.

Tech Tip: Image Sizes

RGB and RGBa images require more space for storage than do grayscale or palette images. Each pixel of an RGB or RGBa image requires four bytes for actual color information and one byte to store information about selection, saved selections and invisible pixel status. At five bytes per pixel a 16000 x 16000 RGBa image requires 1.25 gigabytes of memory. Such a large image, when used for display only, should be converted into a compressed image for much faster speed.
At times when we would like to edit an image we cannot use compressed images and must convert such images back into RGB or RGBA or other image types. In such cases the compressed image will expand dramatically in size as it is decompressed and will require massively more processor time for computation. Working with very large, non-compressed images for anything other than display requires exceptionally fast computers and gigabytes of RAM.
Compressed Images

Compressed images are a special type of image that can be displayed but not edited or otherwise manipulated. In exchange for a limit on functionality, compressed images deliver very fast display capability even when the images involved are very large.

Manifold supports use of compressed images using industry-standard ECW format, the same as used with ER Mapper, or using industry-standard JPEG2000 format. The technology in both ECW and JPEG2000 allows astonishingly fast loading and display of large images (especially with ECW) but this speed is achieved by trading off the benefits achieved against significant limitations. Compressed images may be kept within the project, linked from an external ECW or JPEG2000 file, or linked from an external ECWP server.

The limitations of compressed images are:

- **Display only** - Compressed images may be viewed but not manipulated. They are read-only.
- **No selection** - Compressed images cannot be altered and their pixels may not be selected.
- **Limited on-the-fly re-projection** - A compressed image may be viewed only within a map whose coordinate system of which is more or less similar to that of the image. If the re-projection between the coordinate system of a compressed image and the coordinate system of a map containing the image is curvilinear, the compressed image layer will not be displayed in the map.
- **Reduced information content** - The compression process eliminates the original pixel information in the image. Although compressed images when reconstituted for display may appear visually identical to an original uncompressed image, the reconstituted pixels are a synthetic approximation of the originals.
- **No transparent pixels** - ECW and JPEG2000 compression do not support four channel (RGBa) data so there is no channel available to save transparent pixel information. Note that when working with other types of images and “deleting” pixels the pixels are not really deleted - they are simply made transparent. The only way to delete pixels in an image is to crop the image, in which case the cropped pixels are genuinely deleted. If images containing transparent pixels are converted to a compressed image the transparent pixels will be restored as black regions.

Compressed images nonetheless provide significant benefits:

- **Fast display** - The reconstitution process provides for very rapid display of an image view at any desired zoom or pan.
- **Fast loads and saves** - Projects containing compressed images load and save much faster than projects using uncompressed images.
- **Linked images** - Compressed images may be left in their original .ecw files or in their JPEG2000 files instead of being imported into the Manifold .map project file.
- **Reduced disk space** - Compressed images require much less room than uncompressed images. Using linked images to utilize a single image file in many different projects will also save much disk space as compared to duplicating that image within many projects as an imported image.
- **Streaming links** - Compressed images in ECW format may be linked into a Manifold project from an ECWP server URL, loading from a data stream served by an ER Mapper Image Web Server.
- **Dynamic channel combinations** - The **Image - Display Options** command allows selection of which channels from a multi-channel image will be used for R, G, B and alpha channels.

Compressed images are a good choice for images displayed for their visual merit only, such as when images are used to form a background layer for a map, or when a background image layer will be traced to create a new vector map.

Any image type within Manifold may be converted into a compressed image, and any compressed image may be converted into other image types using the **Image - Convert to** dialog.

**To create a compressed image:**

1. Open the image that is to be converted to a compressed image.
2. Choose **Image - Convert to**
3. In the **Convert To** dialog, choose **Compressed** as the **Target** format. Choose the desired **Ratio** of compression. Larger **Ratio** numbers will result in greater compression at the cost of lower image quality.
4. If desired, check **Save result as new component** to save the result as a new image component instead of converting the opened image. Press **OK**.

The **Ratio** value specifies how large the compressed image is relative to the original image. A ratio of 10 will create a compressed image that is one-tenth the size of the original image.

**To import a compressed image:**

1. Choose **File - Import - Image**.
2. In the **Open** dialog choose **ECW / JPEG2000 Files** in the **Files of type** box.
3. Browse over to the desired .ecw or JPEG2000 file desired and open it.

Importing any .ecw or JPEG2000 file will automatically create a compressed image.

**To link a compressed image from an ECW or JPEG2000 file:**

1. Choose **File - Link - Image**.
2. In the **Open** dialog choose **ECW / JPEG2000 Files** in the **Files of type** box.
3. Browse over to the .ecw file or JPEG2000 file desired and open it.

Linking any .ecw file or JPEG2000 file will automatically create a compressed image referring to a linked file. Working with the image will fetch image data on the fly.

**To link a compressed image from an ECWP server:**

1. Choose **File - Link - Image**.
2. In the **Open** dialog choose **ECWP Image Servers ()** in the **Files of type** box.
3. In the resulting **Link ECWP Data** dialog, provide the URL for an ECWP image server in the **Server** box. The **Server** box remembers most recently used ECWP servers, including those used in previous Manifold sessions.
4. If desired, press the **Test** button to verify the connection. Press **OK**.

Linking to an ECWP image server will automatically create a compressed image referring to that server. Working with the image will fetch image data on the fly.

**To export a compressed image:**

1. Choose **File - Export - Image**.
2. In the **Export** dialog choose **ECW / JPEG2000 Files** in the **Files of type** box.
3. Enter a name for the file in the **File name** box, using .ecw extension if an ECW file is desired and a .j2k extension if JPEG2000 is desired and press **Save**.

Compressed images can only be exported into .ecw files if the benefits of the compression technology are to be retained. Exporting into any other image format will lose the ability to do dynamic compression. Although some file formats, such as .jpg, do use compression they do not allow dynamic, multi-resolution compression as does .ecw format.

**Limitations on Re-Projection**

Except for limited exceptions of interest only to experts, compressed images, linked images and image libraries cannot in general be re-projected.

For example, opening an ECW compressed image and attempting to re-project it to a different projection or dragging and dropping either an ECW compressed image or an image linked from some image server into a map that uses a different projection will fail. In such cases, Manifold will pop open a dialog telling us of the projection incompatibility.
If we would like to re-project such images, we should first convert them to an unlinked local image (in the case of linked images) or to an uncompressed image type (in the case of compressed images) or to a regular local image (in the case of an image library). We can then re-project the image as desired.

Note by the way that attempting to re-project such images on the fly by dropping them into a map that uses a different projection usually indicates a conceptual error on the part of the user: usually, large images of the sort that are used as compressed images, linked images or image libraries are likely to be the largest, or among the largest, layers in any such map. It is therefore would be wisest to use their native projection as the projection for the map, so that when the map is displayed it is other, smaller layers that must be re-projected on the fly to display the map and not the large image layer.

To do so, create the map using the image layer first. This will assure that the map uses the image's projection. Next, drag and drop the other layers into the map. For maximum speed, re-project the other layers to match the projection used by the map. This is easily accomplished by (in the map window) right-clicking on the layer's tab and choosing Project to Map from the context menu.

**About Compressed Images**

Any grayscale, palette, RGB or RGBa image is a *static* image in that it consists of a fixed number of pixels where each pixel consists of an unchanging channel value or values. Every time the image is opened the values will be exactly the same.

Compressed images are *dynamic* images that are created from static images by a compression algorithm that reads the fixed pattern of pixels and encodes the apparent visual content of those pixels as seen by humans into a reduced amount of data. Whenever we wish to view the dynamic image, the data representing it is fetched and used by a special algorithm to reconstitute an image with the same approximate visual appearance as the original static image.

The compression and dynamic reconstitution process allows very large images to be saved in a much smaller amount of disk space and allows very fast viewing. The reconstitution algorithm fetches only that part of the compressed data that is necessary to reconstitute an acceptable visual effect at whatever zoom level (resolution) we wish to view the image. When creating a compressed image, Manifold allows us to choose what level of compression we want. Higher levels of compression will result in images that have less visual fidelity to the original.

Consider the RGB image seen at high zoom level above. We have zoomed in so that individual pixels are visible. The scene illustrated is the Northern part of Baja California in Mexico and the Sea of Cortez.
After conversion to a compressed image, the scene at the same zoom level appears above. Note that individual pixels are no longer discernable because the compression technology used synthesizes a scene using full screen resolution to utilize however many pixels are available in the computer monitor's display.

Although the scene looks very realistic (considering the high zoom into what was originally a low-resolution image), it is very important for scientific purposes to understand that the information content in the compressed image is less than that of the original. The compressed image only looks good visually because a clever decompression-on-the-fly algorithm was used to synthesize a visually appealing image from a smaller number of bytes.

The compressed image may appear to be the same but the data content of compressed images is not fixed as with non-compressed images. Non-compressed images have fixed, constant values for their pixel channels while compressed images have variable results depending upon the scale at which they are displayed and the variation caused by the compression/decompression process. If the content of images contains some specific data, such as representations of heights or specific measurements by instruments at a given location, compressed images will not preserve that data exactly. At best, they will provide an interpolation that approximates the original data.

For this reason, compressed images should not be used to store image data if that data will subsequently be subjected to analyses in which the original value of each pixel must be preserved. For example, some satellite sensors may measure reflectance and save the specific reflectance values in the R, G and B channels of an image, which will then later be analyzed to determine the material constituents of the view (such as snow, ice, sand, water, etc.) based upon reflectance of each pixel. Such an image should not be stored as a compressed image because the compression and decompression process will alter the stored values from the original measurements made by the satellite sensor.

Compressed images are best suited for visual display. The compression / decompression process is a very reasonable compromise considering the perceptual interpolation done by the human visual system. In fact, the results of the process often will give compressed images a "better" appearance to the human eye than the original uncompressed image.

Many tasks in GIS involve the use of images purely for display. Using images as backgrounds in maps with drawing layers overhead is a part of very many tasks ranging from the display of various maps to use of tracing to create new drawings. Compressed images are ideal for such purposes.

**Importing and Linking Images**

Compressed images may be either imported into a Manifold project or they may be left outside the project and linked into it. Compressed images may be linked into a Manifold project either from an .ecw file, a JPEG2000 file or from an ECWP image server.

Compressed images that are linked into a project will be shown in the project pane using a linked image icon.

To link a compressed image into a Manifold project from a file:

1. Choose File - Link - Image
2. In the Files of type box choose ECW / JPEG2000 Files.
3. Browse over to the .ecw file or JPEG2000 file to be linked and open it (either by double-clicking on it or by clicking on it and choosing Open).

Linking a compressed image from an ECWP image server is similar.

To link a compressed image into a Manifold project from an ECWP server:

1. Choose File - Link - Image
2. In the Files of type box choose ECWP Image Servers (
3. In the Link ECWP Data dialog that pops open, specify the URL to a server.
4. To test connectivity to the server without commanding a link, press the Test button. If a connection can be correctly established, a Connection successful message box will appear. Press OK to close the message box.
5. In the Link ECWP Data dialog, press the OK button.
6. A linked image icon will appear for the data from the image server.

Maintaining compressed images in external .ecw files provides several advantages:

- **Fast project loading** - Projects containing linked compressed images will open very rapidly because they will be relatively small in size since the .map project file does not contain any storage space for the linked images.
- **Fast image loading** - Compressed images loaded from disk files will load just as fast as if they were stored inside the project.
- **Reduced disk space** - If several projects use the same compressed image only one copy of the image need be kept on disk.

Linking compressed images into a project has these disadvantages:

- **Multiple files** - If parts of a project are stored outside the .map project file one can no longer send a single .map file to a colleague and be sure that all data is included. All .ecw files used in the project must also be sent. In addition, we must be careful not to delete or move .ecw files that are used in projects.
- **Network delays** - If image files are linked over a network or when linking image files from ECWP servers through a network image display speed may be dramatically reduced due to delays caused by relatively slow network speeds compared to processor - disk speeds in a local system. This will not be a factor on gigabyte local area networks but may be a significant factor if image files are linked through Internet connections.

If a linked image cannot be found (perhaps because the project file has been moved), when the compressed image is opened an empty image window will result. To reconnect the image, right click on the linked image in the project pane and choose Relink, browsing over to the new location of the image.

**Preserving Paths to Linked Images**

Linked images are saved within Manifold projects using relative pathnames. Moving a .map file that contains a linked image will preserve the link to the image provided that the relative path from the .map file to the linked image file has not changed.

For example, suppose we have a .map file located in C:\projects\Carson\mymap.map that contains a linked image located in the file C:\projects\Carson\images\airport.ecw. If we move the .map file to D:\backups\January\mymap.map it will still preserve the link to the image if the image file is moved to D:\backups\January\images\airport.ecw.

**Unlinking a Linked Image**

Compressed images that are linked into a project may be converted into an imported image at any time.

To convert a linked image into an image within the project:
1. Right click on the linked compressed image in the project pane.
2. In the context menu choose **Unlink**.

Unlinking a compressed image linked from a file will import that image into the project as a compressed image stored within the project.

Unlinking a compressed image linked from an ECWP image server will import that image into the project as an RGB image (since necessary wavelet data cannot be fetched through ECWP protocol to construct a local compressed image). Be careful when unlinking an ECWP server image since the resulting RGB image can be a very large image. Once the ECWP server image is unlinked into a local image it can be converted into a compressed image to save space and improve performance.

**Compressed Images and MrSID Format**

If a suitable decoder has been installed, Manifold has the ability to import MrSID format images. See the **Import Image - SID, MrSID** topic for details.

Since MrSID has largely been replaced in modern usage by open formats such as ECW and JPEG2000, Manifold will import MrSID images but it will not export images to MrSID nor utilize MrSID as a native format for linking images as compressed images as is possible with ECW and JPEG2000.

The assumption is that if you are working with Manifold and must deal with an image in MrSID format, your first objective will be to immediately convert the image into a compressed image and to save it out as an **ECW** or **JPEG2000** format image. Once the image has been converted into and saved as a compressed image, loading it and viewing it will be instantaneous and it no longer will be imprisoned in MrSID format.

Because many images are trapped in MrSID format and most GIS users would prefer to use a more open format, some software developers who have licenses to work with both MrSID and ECW or JPEG2000 have created scripts or utilities that can automatically traverse a hard disk and find and convert all MrSID files into ECW files. That is the most efficient way of freeing images trapped in MrSID format since the lengthy process of converting a MrSID image to a more modern format can be accomplished at night or over the weekend. Check the **manifold.net** website's **Free Stuff** page for examples of such scripts.

**See Also**

- **Image - Convert to**
- **Export Image - ECW / JPEG2000**
- **Linked Images**
Intermediate Levels and Pyramids

Manifold is often used to display very large images using technologies such as Compressed images or Image libraries or Linked images served from image servers or linked images from Oracle Spatial that make it feasible to display truly immense images on a desktop computer.

Most such technologies for displaying very large images will include the use of intermediate level images or pyramids to allow much faster display of images, including zooming in or out and panning the image. These two terms mean the same thing, but some software packages will refer to this idea as pyramids while others will use the term intermediate level images or simply intermediate levels. It's the same idea, which may be explained as follows.

Consider a very large image that is shown above in greatly reduced form. The image shows one meter resolution TerraServer photos that have been combined into a single large image that covers the region in the Montara Mountain example surface. See the Combine a Surface and a Drawing in a Map example topic for an example that uses the Montara Mountain sample data.

This image was created by opening the Montara Mountain surface and then linking an image from TerraServer. The drawing was then downloaded at highest resolution using the Image - Download command.

The image shows a region of the San Francisco peninsula ranging from the San Francisco International Airport in the upper right down to the harbor at Half Moon Bay at the lower left. It is over 700 megabytes in size.
If we zoom all the way into this image in the vicinity of the San Francisco airport so that each image pixel occupies one pixel on the computer monitor (so-called Native resolution) we can see that the image is extraordinarily detailed. The image is so detailed that thousands of individual homes and tens of thousands of individual automobiles can be discerned if we zoom all the way into the image.

Computer monitors usually display images using resolutions of about 72 pixels per inch. Even though that results in millions of pixels available in a large monitor, this example image is so large at 11119 x 13929 pixels, that at 72 pixels per inch it would require a computer monitor over 12 feet wide (about four meters) and over 16 feet tall (over five meters) to display the entire image at full resolution on a computer screen.

Obviously, therefore, we will never see this entire image at full resolution on our computer monitor. We will only see parts of the image at full resolution when we zoom in to specific locations. If we zoom far enough out so that the entire image fits, we won't really be looking at the image's pixels, but instead we will be looking at an approximation or interpolation of the image that averages out millions of pixels per square inch to show us an approximate visual representation of what the real image might look like when viewed at greatly reduced scale.

For example, the image above is roughly 1/100th of the size in width and height of the image seen at full resolution. In rough approximation, instead of being 154 inches wide it is only approximately 1.5 inches wide. Therefore, each pixel in the above image represents all of the pixels in a 100 x 100 region of the full resolution image. Each pixel represents over 100 x 100, that is, 10,000 pixels in the full sized image. At 72 pixels per inch,
each square inch of the above greatly reduced view was created by calculating an interpolation of almost 52 million pixels of the full image. Just one square inch of reduced view requires a lot of calculations.

To calculate what the full image would look like at the above greatly reduced view requires computations on hundreds of millions of pixels. That can take a lot of time even on a very fast desktop machine so it is no surprise that truly huge images might appear to be slow whenever we zoom in or out or pan them. Each such step can require a lot of computational power and processor time to compute the necessary view by interpolating millions or hundreds of millions of pixels.

The trick to speeding up that process is make those calculations in advance and to store the results. If we zoom out from the full image so that the entire thing fits on the screen as in the example above the resulting view requires only a few tens of thousands of pixels. It is insignificantly small compared to the hundreds of megabytes of the full sized image. We may as well make the computation just once (perhaps when we store or export the image) and then save the result for use whenever anyone ever wants to see a zoomed out view of the image.

If someone wants to see a zoomed out view, instead of repeating the massive calculation we can simply fetch the pre-built view and display it. Since computer monitors show relatively small numbers of pixels (the number of pixels on even large, high resolution computer monitors is but a tiny fraction of the size of really big images), we can fetch pre-built images to fill up a computer monitor virtually instantaneously. That's a lot faster than computing such zoomed out views on the fly.

The whole idea of intermediate image levels, therefore, is when a very large image is first created or stored our software will automatically compute views of that image at various zoom levels and will store those views along with the image. Programs that display the image can then use those pre-computed, stored views whenever anyone wants to zoom in or out to see different parts of the image at different zoom levels.

In the case of our full resolution example, as seen above, our software might also compute views at regular intervals such as zoomed out at twice the scale, four times the scale and so on.
The image above shows the same region at lower resolution (1/4th the zoom) so that only 1/16th as many pixels are used to cover the same region. A series of images at this intermediate zoom level covering the entire image could be stored with the full sized image.

Zooming even further out requires only 1/64th as many pixels to cover the same region. Another way of saying the same thing is that when zoomed out like this we can use the same number of pixels to show a region 64 times larger.

Obviously, if we compute intermediate levels of zoom and save extra images that takes time. If we compute and store very many intermediate levels of zoom that will also increase the storage size required for images because in addition to the image itself there will be additional space required to store the intermediate level images that are created.

Different technologies have different ways of figuring out what is a reasonable number of views and how to implement storing those intermediate views, but the basic idea is the same: compute some reasonable number of intermediate views and save them so that later on display of the image at different zoom levels will be very fast.

It turns out that with most such technologies it is not a great burden to take the time to compute intermediate levels in advance. That's because such computation usually is done when an image is being stored or exported.
in some process that already will take a significant amount of time. Delays in such cases are not as unpleasant as delays when we zoom in with our mouse and are waiting expectantly for something to happen right away with the image.

In addition, such delays when creating or exporting the image are one-time delays: after the intermediate levels are created once they never again have to be created. We might have a slightly slower, one-time creation process but then every time we view the image our will go much faster.

The need for extra storage space also is not usually a burden, since most people would happily see an image increase in storage space by, say, 50% if thereafter display was virtually instantaneous instead of taking minutes for each change in zoom.

Why are intermediate level images also called pyramids?

If we imagine a sequence of such images stacked up we see that the reduced intermediate levels form a conceptual pyramid of sorts in that "higher" images are smaller than "lower." In addition, when designing such software the data structures used for storage often are diagrammed using hierarchical diagrams that look like pyramids, hence the name.

The bottom line is that whenever we have the choice of using intermediate levels or pyramids, we should always choose that option. It’s a very fast and convenient technology that is well worth the small increase in one-time processing time when creating such images as well as worth being any increase in disk space consumed. Remember, disk space is virtually free while time spent at a keyboard tapping your fingers waiting for a zoom to happen is priceless!

**See Also**

**Image Libraries**
Linked Images

Linked images are images that are stored outside of a Manifold project file and which are dynamically fetched into the project as necessary when the linked image is opened or otherwise used. Note that linked images cannot normally be re-projected (see comments below).

Images may be linked from a variety of external sources, including:

- **Image libraries** - Folders on disk that contain image tiles consisting of images in graphics formats supported by Manifold. For example, a large linked image may be linked in from an image library folder that contains dozens of smaller images in .tif format (the image tiles) that mosaic together to form the larger image. Image libraries are good for smaller collections of images. Larger collections of images or image tiles should be stored within a spatial DBMS (see below) for performance.
- **Compressed images** - Images stored on disk in ECW or JPEG2000 format.
- **ECWP image servers** - Servers providing compressed image data streams using ECW format.
- **Google servers** - Google servers providing images rendering maps or satellite images.
- **Manifold IMS Web Sites** - Manifold IMS can be configured, in addition to standard IMS usage or to OGC WMS image serving, to serve images as an image server.
- **OGC WMS servers** - Servers providing data streams of image data using OGC WMS format.
- **Oracle servers** - Oracle DBMS servers providing Oracle Spatial or Locator facilities within standard Oracle databases to store images.
- **Spatial DBMS servers** - Almost any databases managed by Manifold as spatial DBMS can store images and surfaces in a high performance way.
- **TerraServer** - Servers providing image data in TerraServer format.
- **Tables and Queries** - Linked images may be created dynamically from tables or queries.

Linked images are shown in the project pane using an icon that includes a yellow "database" cylinder to show they are created from some other source. Linked images are normally read-only.

Opening a linked image brings the data for the image in from the source. When the image window is first opened, it will appear with a gray, partially transparent background that will fill in as the image is brought into Manifold. Most linked images are linked to sources that send the image in as tiles, that is, as rectangular pieces that form a seamless mosaic of the complete image. As we zoom in or out of the image, or as we pan in different directions, Manifold will fetch tiles from the image source to create the appropriate view.

As the tiles are fetched from the source, they will fill in more and more of the image window. Manifold tries to fetch tiles that are closest to the center of the visible area of the image and then will fetch tiles that are nearer to the edge of the visible area, in the expectation that we are usually most interested in seeing the center of the image and would like to begin viewing that while the rest of the image fills in.

The speed with which image tiles appear depends on the technology used by the image source as well as by the connection to the image source. For example, a linked image created from an ECW file stored on local disk will be breathtakingly fast even if the image is many gigabytes in size. Likewise, an image linked from a local Oracle server will be amazingly fast. However, an image using a slower technology, such as an image library assembled from .bmp files could be much slower to come into view.

Images linked from remote servers accessed by Internet connections, such as TerraServer or OGC WMS servers cannot be any faster than the Internet connection to that server or the speed of the server. Some OGC WMS servers, for example, are appallingly slow to begin with and are further overloaded with too many users. It can take seemingly forever to get images from such servers.

Manifold will try to help us out in such situations by caching any fetched tiles if the Cache data between sessions option is turned ON (the default) in the image linking dialog. In that case, if we ever return to a view already fetched, such as, for example, by panning to the right and then panning back to the left, Manifold will not need to fetch the necessary tiles a second time: they will be saved in cache and will instantly display.

Another important command used to get around slow image servers is the Image - Download command, which we can use to instruct Manifold to get all tiles for a desired view in background while we continue working on other things.

Linked Images from Tables or Queries
The tables and queries used to create linked images may be inside the Manifold project or they may be in external databases.

Linked images may also be created in a two step process, where the virtual table for an existing image is used in a query to manipulate that image or to fetch part of the image based upon desired criteria. A linked image can then be created from that query. This technique is often used within Manifold projects that will be used in an IMS website.

Linked images may also be created from tables or queries stored in external databases. For example, we could use a query to select all columns for all pixels in an image's virtual table and then export that query as a table into an .mdb file or to some other database storage.

When an image is linked from a table or query, each record in the table or query produces a pixel. The coordinates of the pixel are taken from the X and Y columns (cast to integer) specified in the Link dialog. The color of the pixel is taken from either the Color column or the channel columns.

**Editing Linked Images**

Although a linked image that appears in the project pane is normally read-only, in the case of linked images created from tables or queries it is often possible to edit that linked image by editing the data source from which it is created. For example, suppose we have a linked image created from a query that uses the virtual table of another image. We could change that linked image by editing either the original image or by editing the query.

**Linked Images and IMS**

Linked images are therefore perfect for a wide class of Manifold Internet Map Server applications where images must be dynamically updated based on changing information stored in database management systems. For example, an image providing a background to a map may be taken from an ECWP streaming server that provides a satellite image showing the weather in the region as it exists at the time, or an image showing a weather chart may be fetched from an OGC WMS server that synthesizes an image from a vector representation of weather data.

**Linked Image Tiles are Fetched on Demand**

One especially useful characteristic of images linked from image servers is that the tiles that comprise an particular view are fetched on demand only when that particular scene is viewed. When a user zooms or pans into a particular view (or jumps directly there as a result of a Goto or other navigation command) Manifold contacts the image server and fetches image tiles of the required resolution that are automatically formed into a seamless mosaic for the visual display that fills the image window at the desired zoom at the location specified. If the user zooms further into the image then higher resolution tiles will be fetched to the limit of resolution provided by the image server. Zooming yet further into the image inflates the single, full resolution tile on view to create an ever more pixilated display.

By default, the image tiles used to build requested views of a linked image will be cached on disk by Manifold. If a particular view of a linked image is revisited, such as by using the Back navigation toolbar button, Manifold will redisplay the desired view by fetching necessary tiles from the disk cache rather than from the image server. Such redisplayed views will normally be much faster than previously unseen views since disk access is a much faster process than streaming data through even a fast network.

If portions of a very large linked image, such as a linked image of the entire United States, are never viewed then the image tiles for those portions will never be fetched. This characteristic together with caching allows us to use linked images to great effect in certain applications, such as IMS applications.

Suppose our IMS application is a web site that shows the location of schools in Ohio. We would like to have an image layer in our web site that, when zoomed in, shows aerial photography from an image server so that photographic details of each school, such as the location and nature of athletic fields and other facilities, can be seen. Perhaps the intent of the site is to assist the operation of a state-wide soccer league so that league planners, coaches and parents can get a visual preview of different facilities before scheduling or attending soccer matches.

If the image server we would like to use provides state-wide data, we could create this application by simply creating a linked image for all of Ohio to use as a layer in our web site's map. The initial view when we create our
application would consist of a relatively small number of very low-resolution tiles that form a mosaic to show the entire state.

As visitors to our site navigate to various individual schools (say, by giving the school's street address and going directly to each school, by picking names from drop down lists of counties and then school names or by some other suitably efficient navigation method), they will get portions of the overall Ohio linked image served to them at whatever resolution is required for the zooming and panning they do about each school. Image tiles for each view will be cached on disk.

The schools of greatest interest to visitors will naturally be viewed first and so the image tiles for those schools will be the first to be saved in cache. As a result, views for those schools will be fast and require no connection overhead to the image server, exactly the result we would like in a well-architected web site.

After a while as a result of many views of many schools by many visitors, the server machine will have many image tiles stored in cache on its hard disk. A visit to virtually any school in the system used for soccer games will utilize cached image tiles and will not require download from the image server. The cached image tiles on disk will represent a fraction of image tiles that are available for all of Ohio, clustered as they will be in a patchwork of tiles at the location of each school. But no image tiles will be wasted on high resolution views of vast stretches of farmland or other regions not viewed by visitors to the web site. Since a visit to almost any school will used cached tiles, the web site will be very fast as it will rarely need to fetch new tiles from the image server.

If we wanted to assure even greater efficiency we could apply a zoom range to the linked image layer so that it only appears when visitors to the web site are zoomed far into the map. If they are zoomed far into the map we can safely reckon they are looking at a close-up view of a particular school site and so the linked image layer should be displayed. In that case, should users choose to navigate to a particular school by successively panning and zooming further into the web site map (which presumably shows major streets and towns as an aid to navigation) instead of by jumping directly to a school using whatever clever navigation user interface we have provided, then no image will be displayed as background during the process of finding a particular school. No image will be displayed until the user zooms into a particular school and thus no image tiles will be cached during the process of finding a particular school.

That would be more efficient because we would avoid any delay while low resolution images of parts of Ohio are fetched in order to provide a background visual for such pan and zoom navigation to a particular school, and no space on disk would be consumed for any cached image tiles used in such navigation. It is true that if we did not use zoom ranges over time the server would accumulate in cache all of the low resolution image tiles for Ohio and many of the medium resolution tiles used in such navigation, so this method of navigation would become faster and faster as more tiles would become available from disk cache. It is also true that disk space is nearly free so that most webmasters for such a useful application wouldn’t worry about using a few tens of gigabytes of additional disk space to allow showing a background image all of the time.

However, it does take time to build up a fast cache and a perceptual delay when panning is something that for psychological user interface reasons especially should be avoided. Most people have used Internet so much that they expect when they jump to a new location they may have to wait a few moments to get the display they want, but when they pan through an apparently seamless view of a map or an image they expect fewer delays. A further point is that for large regions the number of tiles saved for various intermediate views could get very large as users take various random paths of panning and successively zooming to find a particular location. Finally, although everyone enjoys a view from space, the visual clutter of a background image is usually not the fastest and most comprehensible way of helping visitors orient themselves. Simple, clear maps, the simpler the better, are usually the best way, with detailed images reserved for close-up views once a location of interest has been found.

Therefore, although we might get away with not using zoom ranges on a linked image of Ohio used in a web site, it is unlikely we would not use zoom ranges if the web site showed the entire United States. Even for a smaller region such as Ohio most webmasters would use zoom ranges to suppress the background image until zoomed far enough into the map to view an individual school.

**Note:** On demand retrieval of tiles as discussed above applies to images linked from Manifold Image Servers, images linked from OGC WMS servers and images linked from TerraServer.

**Relink and Unlink**

If a connection is lost between a linked image and its originating data, the **Image - Relink** command allows us to restore the connection. If we would like to convert a linked image into a regular image, the **Image - Unlink** command will sever all connections to the originating source and will convert it into a regular image.
Converting a Linked Image to a Local Image

An image linked from a remote server can be converted to a native (local) image within the project using the Image - Convert To command.

Limitations on Re-Projection

Except for limited exceptions of interest only to experts, compressed images, linked images and image libraries cannot in general be re-projected. They can only be re-projected into a projection that is affinely equivalent (change of scale and location only) to the projection in which the image exists. As a practical matter, this means that for most purposes such images cannot be re-projected or used in maps that use a projection different from the image.

For example, opening an ECW compressed image and attempting to re-project it to a different projection or dragging and dropping either an ECW compressed image or an image linked from some image server into a map that uses a different projection will fail. In such cases, Manifold will pop open a dialog telling us of the projection incompatibility.

If we would like to re-project such images, we should first convert them to an unlinked local image (in the case of linked images) or to an uncompressed image type (in the case of compressed images) or to a regular local image (in the case of an image library). We can then re-project the image as desired.

Linked Images and Map Projections

If we need to use a linked image in a map, the map should be created from the image so that the map uses the same projection as the image. If we try to include a linked image as a layer within a map that uses a projection different from the linked image, the image will not appear in the map because it cannot be re-projected into the projection used by the map. Manifold will warn us if we attempt to drop a linked image into a map with an incompatible projection.

One way around this projection limitation is to unlink the linked image so that it is converted into a local image. We can then re-project as we please, although of course in this case the dynamic linkage between the image and the source will no longer apply. It will be a static, local image stored in the project just like any ordinary image.

Note by the way that attempting to re-project such images on the fly by dropping them into a map that uses a different projection usually indicates a poor performance choice on the part of the user: usually, large images of the sort that are used as compressed images, linked images or image libraries are likely to be the largest, or among the largest, layers in any such map. It is therefore would be wisest to use their native projection as the projection for the map, so that when the map is displayed it is other, smaller layers that must be re-projected on the fly to display the map and not the large image layer.

To do so, create the map using the image layer first. This will assure that the map uses the image's projection. Next, drag and drop the other layers into the map. For maximum speed, re-project the other layers to match the projection used by the map. This is easily accomplished by (in the map window) right-clicking on the layer's tab and choosing Project to Map from the context menu.

If we attempt to create a map using several linked images at once that use different projections, then at least some of those images will have projections incompatible with that chosen for the map. Manifold will display a warning message if we attempt to do this.

Properties

The View - Properties dialog for components shows the data source of a linked component (such as an image library or other linked image) and other relevant information.

The Link / Share dialog accessed from the [...] browse button for linked or shared components accessed from the View - Properties dialog will provide a summary of the link or share properties. This will show the status of a shared component and whether or not changes in a linked component will propagate back to the data source.

For Enterprise Edition Users
Sharing a linked image to an Enterprise server places the linking information into the Enterprise server and not the actual image data. Importing or linking such a shared linked image to a local project will re-establish the connection to the relevant file or server. See the Enterprise Edition topic for information on Enterprise Edition.

**Tech Tip: Using Linked Images in Print Layouts**

It's not a good idea to use in layouts images that are linked from external servers (such as from ECWP servers, OGC WMS servers or TerraServer servers) because such images are downloaded as necessary to fill a particular view at a given zoom level and pan.

If such images are used in print layouts, it could well be that the tiles necessary to show the image at whatever resolution ends up being necessary for printing at a given printer resolution will not have been downloaded at the time the print job runs. Since it can take a long time to fetch such tiles, the system will use whatever is available for the print job, resulting in odd effects such as images that are blank or partially blank.

A further problem is that the image might change on the server between the time a print layout is created and when it is actually printed.

The way to avoid any unexpected effects is to first download the image to a regular image at the desired resolution and then use the regular image in the print layout.

**Tech Tip: Permissions and Linked Images in IMS Applications**

When using linked images in an IMS web application it is important that the web application has read and write permissions in the cache folder, or otherwise it will not be able to use the tiles cached for the linked image.

**See Also**

- Compressed Images
- Linked Images from Google Servers
- Linked Images from OGC WMS Servers
- Linked Images from Oracle Servers
- Linked Images from TerraServer
- Image Libraries
  - Virtual Tables for Images and Surfaces
  - Queries and Images or Surfaces
  - Linking Images
Image Libraries

An image library component is a type of linked image component that shows what appears to be a single image that is created by automatically forming a mosaic from one or more images saved in image files within a folder on disk. The individual images that comprise the image library are called image tiles. The image library component takes its name from the name of the folder that contains the image tiles that comprise the image library. Image tile files may be in BMP, ECW, GIF, JP2 (JPEG 2000), JPG, PNG, TGA or TIFF formats.

- If the image tile files are in ECW, JPEG 2000 or GeoTIFF formats that contain projection information within the file then the coordinate system information within each image file will automatically be used to georegister the image and form the image library mosaic.
- If the image tile files are accompanied by like-named files containing projection information in XML, PRJ, GSR or world files then the coordinate system information from those accompanying files will automatically be used to georegister the images and form the image library mosaic.
- If the image tile files are not accompanied by like-named files containing projection information, then the image library mosaic can still be created if the image tile files are named to indicate their position in the mosaic.

We can link an image library into a project by choosing File - Link - Image and then choosing Image Library Files in the Files of type box and then navigating into the folder in which the image library images are located and choosing one of the images. This launches the Link Image Library dialog.

Link Image Library Dialog Controls

- **Arrange images using filenames** If checked (the default), derive the location of image tiles from the name of each image file and place images accordingly. Choosing this option ignores coordinate system data in any accompanying coordinate system files or in the image files themselves.
- **Mask** A template using \[X\], \[Y\] and optionally \[L\] nomenclature to show how the names of image files should be used to place each image tile into position.
- **Automatically save intermediate levels** If checked (the default), will automatically generate tiles for intermediate image levels and will store them into the folder together with the other image files.
- **Reverse Y** Reverse the order in which \(Y\) values are interpreted.
- **Margins** The number of pixels to clip from left, right, top and bottom margins. Used when image tiles overlap and some part is to be deleted.
- **Georegister images using accompanying files** If checked, derive the location of image tiles from accompanying coordinate system files or from the files themselves for supported formats.
- **Mask** The template used for a filename when the above option is checked. Normally just an asterisk and the extension for that type of image file, such as .jpg, .tif or other supported image format.
- **Ignore datum differences** Enabled when the coordinate system for each image is to be defined by an accompanying file. If more than one datum is in use the system will arbitrarily choose one such datum from those used and will then treat all datums as if only that arbitrarily selected datum was the controlling datum.
- **Scan subfolders** If checked (the default), scan folders recursively below this folder to find more image files. If not checked, will only use image files from this folder.

Creating Image Libraries using Georeferenced Image Tiles

It's very easy to create an image library component if the image tile files we would like to use are either:
- In GeoTIFF, JPEG 2000 or ECW format and containing georeferencing information (although unusual, it is possible some ECW or JPEG 2000 files might have been created without embedded georeferencing information), or
- Are accompanied by accessory files giving projection information in XML, PRJ or GSR files or are accompanied by world files that give shifts and scales. If accompanying XML files are used they must be the .xml files written by Manifold System to specify the projection / georegistration information for an image.

If image library image tile files fall into one of the above categories it doesn't matter what file names are used or what the size of each image file may be.

To link an image library into a Manifold project using image tiles with accessory projection files.

2. In the Files of type box choose Image Library Files.
3. Browse into the folder containing the image tile files. All image tile files to be used must be in that same folder.
4. Double-click any one of the image files.
5. In the resulting Link Image Library dialog, check the Georegister images using accompanying files option.
6. In the Mask box, provide a mask that matches the file type to be used, for example, *.tif if using .tif files and *.jpg if using .jpg files.
7. Press OK.

A new linked image library component will appear in the Manifold project, taking its name from the name of the folder that contains the image tile files. Opening the image library component will show a gray background that will begin populating with images as the individual image tiles are loaded.

Example

Consider a series of nine images saved in ordinary .tif files (that is, not GeoTIFF files but plain, ordinary tif files as might be processed by PhotoShop and saved without georeferencing tags) in a folder called Bay image. These images may be arranged in a mosaic to form an overall view of the San Francisco Bay area.
Each image is georegistered and was saved (using Manifold) so that it has an accompanying .xml file giving relevant coordinate system information. Each image when imported therefore will be automatically georegistered. The images are all different sizes.

We can create an image library component from these images by choosing File - Link - Image, choosing Image Library Files in the Files of type box and then browsing into the Bay image folder and double-clicking one of the .tif image names. In the resulting dialog, we check the Load coordinate systems... option and in the Mask box we use *.tif as the mask. Press OK.

When we open the resulting Bay image image library component Manifold will import all of the image tiles and use the accompanying .xml coordinate system file for each to know where each file should be positioned to make a combined mosaic that appears to be a single image.

In this example, each image tile neatly abuts the adjacent image with no overlaps or gaps. If any image tiles overlapped, the resulting image library would average the pixel colors in the region of overlap to create the pixels in that region for the image library composite. If there were gaps or no image tile in a particular region, the image library would have transparent pixels in that region.

Image libraries made up from image tile files that are accompanied by XML, PRJ or GRS files that provide exact coordinate system information for each image tile are very easy to work with because the placement of each image is completely automated by Manifold. There's really no need for any significant thought on the part of the user because the files themselves contain all information necessary to mosaic them together. The files can be named using whatever names they have and they can be different sized as well. Easy!

It's also easy to work with GeoTIFF, ECW, or JPEG 2000 files that contain georegistration information inside the files. In that case projection information comes from the files themselves.

We can also create image libraries from image files that are accompanied by world files. World files are a highly imperfect way of storing partial projection information, but they are better than nothing. See the discussion of world files in the Importing Images topic.

Specifying Image Locations using Image File Names

Creating an image library requires more effort when image tile files do not use a geographically aware format like GeoTIFF or ECW or when they do not have accompanying XML, PRJ or GRS files to enable automated mosaics. It is still possible to automatically mosaic such image tiles into an image library component, but we will have to organize the images by naming the files in a regular way so that Manifold can tell from the name of the image where to put each image tile.

Giving images names where the filename indicates where each image is located is nothing new; that's a common technique often used when saving images in various applications so that later on it is easy to tell from file names on disk the geographic area covered by a particular image.

To link an image library into a Manifold project using image tiles without accessory projection files.
2. In the Files of type box choose Image Library Files.
3. Browse into the folder containing the image tile files. All image tile files to be used must be in that same folder. All images must be the same size in terms of their height and width in pixels.
4. Double-click any one of the image files.
5. In the resulting Link Image Library dialog, check the Arrange images using filenames option.
6. In the Mask box, provide a mask that describes the style of file name used to specify the position of the image.
7. If parts of each image should be masked off (for example, to remove some unwanted common border or region of overlap) use the Margins settings to specify the region on each side to be removed.
8. Press OK.

With the above technique we must understand how the naming convention being used should be specified in the Mask box. Manifold provides flexible options so that different forms of file names may be accommodated.

**Example**

Consider a set of images similar to those used in the example above, which may be arranged to form a mosaic of the San Francisco Bay area. Each image is saved in an ordinary .tif file without any projection information.

![Example Image](image.png)

However, in this case, each of the individual images forming the mosaic are exactly the same size in terms of numbers of pixels in width and breadth.
Images on the edge of the mosaic contain regions of black pixels in areas not represented in the original photograph. The images are saved in files that do not have accompanying coordinate system information. Upon import into Manifold, they will not be georegistered.

A collection of images like those above may be used in Manifold to form an image library if they are named in an orderly fashion so that Manifold knows from the file names for each image tile where that image tile is to be placed within the image library mosaic.

We can use a naming scheme that contains within the filename an indication of which X,Y position each particular image tile occupies within the image library mosaic. The conventional way to designate X and Y locations is to begin in the lower left hand corner and to number to the right for X values and upwards for Y values.
If we do this for the mosaic as seen above, we can provide an X,Y coordinate for each image tile. For example, the 3,2 tile is the one located in the third X position in the second Y position, that is, on the rightmost part of the middle row.

We can give each image tile file a name that incorporates the X,Y location of the tile in its name. For example, the image tile that is supposed to go into the 3,2 position will be named `Bay0302.tif`.

Why the use of leading zeroes as well as the name `Bay` within each file name? That's not required but it does help to keep things organized and easily legible. We could have just as easily named the 3,2 tile `32.tif` or
Manifold’s Mask specification used in the image library dialog allows variations like those, as we will see below.

We will give each tile a name using the above scheme. If we were to open the individual images in a package like Photoshop we’d see that they are just a group of unregistered images that happen to have been given names that indicate where they should be placed if a mosaic of the entire Bay area is to be constructed.

If we create an image library component using the above images, we would use a Mask value of

\[
\text{Bay}[X:2][Y:2].tif
\]

to indicate the names of the image tile files. The \([X:n]\) and \([Y:n]\) escape sequences tell Manifold what part of the file name has to do with X and Y coordinates. The \([X:2]\) specification tells Manifold that two characters in that position in the file name specify the X position of the tile in the mosaic and the \([Y:2]\) specification tells Manifold that two characters in that position specify the Y position.

The above mask matches file name specifications ranging from Bay0101.tif to Bay9999.tif. It would not match Bay123ab.tif. Other masks and examples include:

<table>
<thead>
<tr>
<th>Example</th>
<th>Mask</th>
</tr>
</thead>
<tbody>
<tr>
<td>32.tif</td>
<td>[X:1][Y:1].tif</td>
</tr>
<tr>
<td>003002.tif</td>
<td>[X:3][Y:3].tif</td>
</tr>
<tr>
<td>thisoneisthe32tile.tif</td>
<td>thisoneisthe[X:1][Y:1]tile.tif</td>
</tr>
<tr>
<td>img_x3y2.tif</td>
<td>img_x[X]y[Y].tif</td>
</tr>
</tbody>
</table>
When the above mask is used to create an image library component, the individual image tiles will be placed into a mosaic based upon their file names to create an overall image of the Bay area.

**Reverse Y Option**

While it is virtually universal practice to number the X axis with increasing values to the right, not everyone numbers their Y locations counting up from the bottom left corner. Some people and systems prefer to begin at the top left corner and count down.
We can use this arrangement if we like, as seen above, and create different file names. In this case, the Bay0101.tif image is the upper left corner image and not the lower left corner image. If we prefer to name our image tile files in this manner we can still use the same Mask specification. We simply check the Reverse Y option in the Link Image Library dialog to tell Manifold we are numbering Y values from the top down instead of the conventional bottom up.

**Exporting Image Libraries**

An image library may be exported to a file using the File - Export - Image command. The image library will be exported as a single image using the specified image format.

**Unlinking Image Libraries**

An image library may be transformed into an ordinary image by right clicking the image library in the project pane and choosing Unlink.

**Converting Image Libraries**

Another way to convert an image library into an ordinary image is to open the image library in a window and then choosing Image - Convert. Converting an image library creates a local image in the project of the desired type, being equivalent to first unlinking it and then converting the new local image into an image of the desired type.

**Limitations on Re-Projection**

Except for limited exceptions of interest only to experts, compressed images, linked images and image libraries cannot in general be re-projected.

For example, opening an ECW compressed image and attempting to re-project it to a different projection or dragging and dropping either an ECW compressed image or an image linked from some image server into a map that uses a different projection will fail. In such cases, Manifold will pop open a dialog telling us of the projection incompatibility.

If we would like to re-project such images, we should first convert them to an unlinked local image (in the case of linked images) or to an uncompressed image type (in the case of compressed images) or to a regular local image (in the case of an image library). We can then re-project the image as desired.

Note by the way that attempting to re-project such images on the fly by dropping them into a map that uses a different projection usually indicates a conceptual error on the part of the user: usually, large images of the sort that are used as compressed images, linked images or image libraries are likely to be the largest, or among the largest, layers in any such map. It is therefore would be wisest to use their native projection as the projection for the map, so that when the map is displayed it is other, smaller layers that must be re-projected on the fly to display the map and not the large image layer.

To do so, create the map using the image layer first. This will assure that the map uses the image's projection. Next, drag and drop the other layers into the map. For maximum speed, re-project the other layers to match the projection used by the map. This is easily accomplished by (in the map window) right-clicking on the layer's tab and choosing Project to Map from the context menu.

**Index Drawings**

When an image library is open the Image - Create Index Drawing command is enabled. This command allows us to automatically create a drawing that contains areas showing the extent of each individual image file that makes up that image library together with desired information about each image file, such as the name of the file, the path to that file's location and the type of file.

Index drawings are used in a variety of applications. For example, we might not wish to display all of the images in an image library but need to fetch an image that is part of the library. An IMS application, for example, might show a variety of vector layers and use an index drawing to show where imagery is available in a particular region. The application could be programmed so that a user who wants to see that imagery can double-click onto an index drawing area to pop open the corresponding image in a new window or to make it visible in a map window.
See the Image - Create Index Drawing topic for an example that creates an index drawing for the image library seen in this topic and then uses that index drawing within a Manifold IMS web page.

**Intermediate Levels**

If we provide intermediate image levels manually using the [L] specification and user-supplied intermediate level files we can improve interactive performance. See the Intermediate Levels and Pyramids topic for an explanation of intermediate level images, with illustrations.

If we do not manually provide intermediate level files (it's rare that they are available) it is a good idea to keep the **Automatically save intermediate levels** option checked so that Manifold will create intermediate level images for us. Even if we manually provide intermediate level images it is not a bad idea to have Manifold supplement them with automatically created intermediate level images. Here's why:

Most image libraries make up an image mosaic that is far larger than can fit into a computer monitor screen at one to one scale (one pixel in the image appearing in one pixel in the monitor). When we zoom far out enough to see the entire image, we don't see every pixel in the screen: instead, we see an interpolation or averaged out view where each pixel in the zoomed out image represents potentially hundreds of pixels at full resolution.

Computing such averaged out views takes time. The process can be speeded up a lot if intermediate level images, also known as pyramids, are computed in advance for intermediate zoom levels. Saving such intermediate zoom level images doesn't take much space but if they are available when we zoom in or out Manifold can simply utilize a pre-computed view for faster zooms and panning. This is much faster than re-computing an interpolated view on demand.

Most technologies for fast viewing of very large images, such as ECW, incorporate some scheme for pre-computing intermediate views in advance and then fetching them as needed to support faster zooming and panning. The **Automatically save intermediate levels** option allows us to utilize intermediate image levels even when working with image libraries that do not have intermediate levels. When the option is turned on, Manifold will automatically create intermediate level images and save them into the folder being used with the rest of the image library tiles.

**Note:** to use this option, our user login must have write permissions to the folder being used for image library tiles. If our login does not have write permission, Manifold won't be able to write intermediate level image files into that folder.

If the option is turned off, the image library will try to keep autogenerated intermediate image tiles in RAM cache. If the amount of available RAM decreases below an internally-computed safety level, the intermediate image tiles in RAM will be discarded and then regenerated again on demand.

**[L] Level Specification**

The [L] specification allows us to manually specify intermediate level image files to be used in the image library if we have intermediate level files available.

It is often the case that a particular region has images of greater and lesser resolution that cover it. For example, a region might be covered by sixteen images at high resolution and also be covered by four images at lower resolution and by one image at very low resolution. The different sets of images covering the same region represent **intermediate levels** of resolution.

Manifold can exploit the different levels of images to select the best level to match a given zoom into an image library. When zoomed far into an image library the higher resolution level images will be used. When zoomed out, so that very detailed pixels could not be seen anyway, Manifold can use a lower resolution image level for faster display.

The [L] specification in an image filename mask allows us to tell Manifold which images should be used at which level of zoom.

To specify the level to which different image files belong, we use the [L] specification within an image mask. Level number always uses **zero based** counting as in 0, 1, 2, 3... with level 0 being the highest resolution images covering the smallest regions and higher level numbers referring to lower resolution images that cover a larger...
The [L] specification is used together with the [X] and [Y] specifications to describe the organization implied by filenames.

<table>
<thead>
<tr>
<th>Example</th>
<th>Mask</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>132.tif</td>
<td>[L:1][X:1][Y:1].tif</td>
<td>An image tile that belongs to level 1 where levels are specified using one digit.</td>
</tr>
<tr>
<td>01003002.tif</td>
<td>[L:2][X:3][Y:3].tif</td>
<td>An image tile that belongs to level 1 where levels are specified using two digits.</td>
</tr>
<tr>
<td>level3position32.tif</td>
<td>level[L]position[X:1][Y:1].tif</td>
<td>An image tile that belongs to level 3 where levels are specified using one digit.</td>
</tr>
</tbody>
</table>

It is easy to get confused as to what intermediate level image relates to which higher-resolution (smaller) images it covers. We can help keep things straight by using an orderly naming scheme which in the case of image libraries that employ intermediate levels should use zero based numbering.

When numbering X and Y, most people begin with 1 as illustrated above so that the first, corner tile is in the 1,1 position. However, if we like we can number beginning with zero in the series 0, 1, 2, 3... so that the first tile is in the 0,0 position. That means names such as Bay0000.tif for the first corner tile instead of Bay0101.tif are perfectly acceptable to Manifold.

Using zero for the first thing when counting is a bit atypical, though, as most people count their fingers (or anything else) using “one,” “two,” “three” and so on and do not count their fingers using “zero” for the first finger. Although our everyday experience of counting things such as our fingers argues against starting with “zero” for the first item, there is a very good reason why some programming languages and numbering schemes do so: that is to preserve the ability to do modulo arithmetic. In the case of image libraries if we number from zero we make it easier to retain a common naming scheme if intermediate levels are used.

The reason is that intermediate level images will be arranged by their names which, to facilitate a modulo arithmetically related connection to lower level images are always zero based. That is a mask of level[L]position[X:1][Y:1].tif will mean that a name of level2position00.tif is the corner second level image.

Notes

If we have Enterprise Edition installed, image libraries may be shared on an Enterprise server. Note that the image tiles are not actually stored on the Enterprise server: instead, the image tiles remain in the folder where they are located.

If image tile files are accompanied by more than one different type of accessory projection file, then any XML files will be read first and other projection files ignored, next any PRJ files will be read, then any GSR files and finally any world files will be read. If image tile files are not accompanied by any projection files, or if the accompanying files are not in the correct format, the system will use coordinate system data embedded within the image files themselves if the image files are GeoTIFF, ECW or JPEG 2000 files.

Image libraries have many uses. They are especially handy when converting large numbers of legacy image files into a modern format such as ECW: assemble an image library from the many small legacy image files, unlink the image library to create a single image and then export the image as an ECW.

Manifold will use more than one thread to render image libraries if more than one processor or processor core is available on the computer system. Therefore, image libraries will render faster on multiprocessor or on multi-core processor systems. For example, using a dual-core processor will allow image libraries to render faster.

Tech Tip for Experts: Level Nomenclature Illustrated

If you never plan on manually using intermediate levels, you can safely skip this part of the topic. Use the Automatically save intermediate levels option and Manifold will create intermediate levels for you automatically.
and take care of all naming details. This discussion is intended for experts who will be manually working with intermediate levels.

Even for experts, the naming scheme used for levels can be confusing. It’s best to use some illustrations to sort things out.

Consider an image library created from 25 image tiles. For this simple example, each image is 100 x 100 pixels in size and contains a solid green color. The illustration above shows the image tiles forming an image library with borders showing the edges of each tile.

We’ve now added labels that show the name used for each tile. The tiles were arranged into an image library using the **Arrange images using filenames** option. The **mask** used was

\[ L[L]_{[X:1][Y:1]}.tif \]

...with zero-based numbering so that a filename such as **L0_20.tif** means the file is the third X position over from the origin in the lower left, and is in the first Y row. The number 0 after the **L** in the filename means this tile belongs to the zero level, that is the first and highest resolution set of images.
To keep things simple, we will change the labels in the above illustration to simple X:Y representation so we can see more directly how the X and Y numbering scheme works. So far, so good.

Now let's show another layer of images that represent an intermediate level of images.

Shown in blue we see four images, also using the same mask of

L[L]_{[X:1]}_{[Y:1]}.tif

...as a template for their file names. There are only four blue tiles, not quite enough to cover all of the green tiles. This set of illustrations deliberately does not use enough blue tiles to cover all of the green ones so that the illustrations can show both some uncovered as well as covered lower level tiles. Note that all of the blue image tile names begin with an L1 to indicate they are on the next level up from the zero level.

A nuance: it is easy to mistakenly think that a single intermediate level image must be 200 x 200 pixels in size if it covers four more detailed images which are 100 x 100 pixels in size. That's not true. The intermediate level image is also only 100 x 100 pixels in size, but when it is displayed it is automatically scaled (making its pixels "larger" or "smaller") so that it covers the same geographic region as four more detailed images. It covers four times as much ground using the same number of pixels, hence it is less detailed because one pixel in the
intermediate level image cannot convey the same level of detail that four pixels in the more detailed level show. That doesn't coarsen the visual display because the intermediate level image is only used when the image is zoomed far out enough that the greater number of pixels in the more detailed images would in any event go to waste.

Let us now look at the numbering of these blue intermediate level images together with the numbering of the green images.

To make that easier to see, we will change the labels used on the blue images to $X:Y$ labels as well, and we will move the labels on the blue image tiles from the center of the tile down to the lower left corner of the tile.

The above image shows the green tiles with their name numbering scheme indicated in $X:Y$ form as labels in the center of the tile. The blue tiles also have their name numbering indicated in $X:Y$ form in the lower left corner of the tile. There are two things we can see from this comparison:

- First, both the blue tiles and the green tiles use the same mask, which provides a pattern for how each tile should be named based on its position in the image library. The same pattern applies to both the green and the blue tiles. In both cases, the leftmost lowest tile (at the origin) uses 0:0 for its naming. The three tiles adjacent to the leftmost lowest tile in both cases are named 0:1, 1:0, and 0:0.

- Second, we can see why, as a result of modulo arithmetic (that is, integer arithmetic discarding any remainder), it makes sense to use zero based numbering when levels are involved. If we use zero based numbering we can easily calculate for each lower level tile what numbering name should apply to any higher level tile that covers it.

In our above example, each blue tile is twice ($2$ times) the size in $X$ and $Y$ of each green tile. Consider the green tile at the 3:2 position. What should the numbering name be of the blue tile that covers it?

To get the $X$ part of the name we divide 3 by 2 and get 1 (three divided by two to get an integer without any remainder or decimal fraction is one). So the name will be 1:something. To get the $Y$ part of the name we divide 2 by 2 and also get 1. So the name should be 1:1 and, indeed, 1:1 is the name of the blue tile covering the 3:2 green tile.

If we repeat the above arithmetic for every green tile we can see that it works in all cases to accurately predict the name of the blue tile that covers that green tile. If we extrapolate how any additional blue tiles would be named going to the right and upwards we see that the rule continues to work.

This rule may seem to be almost idiotically simple, but it not only can simplify programming that works with image libraries, it can also help save us from simple mistakes. It is such a strong and useful rule that level names always use zero based numbering.
We must keep that in mind to avoid conceptual errors. Let’s consider an example of a possible conceptual error.

<table>
<thead>
<tr>
<th>0:0</th>
<th>0:1</th>
<th>0:2</th>
<th>0:3</th>
<th>0:4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1:0</td>
<td>1:1</td>
<td>1:2</td>
<td>1:3</td>
<td>1:4</td>
</tr>
<tr>
<td>2:0</td>
<td>2:1</td>
<td>2:2</td>
<td>2:3</td>
<td>2:4</td>
</tr>
<tr>
<td>3:0</td>
<td>3:1</td>
<td>3:2</td>
<td>3:3</td>
<td>3:4</td>
</tr>
<tr>
<td>4:0</td>
<td>4:1</td>
<td>4:2</td>
<td>4:3</td>
<td>4:4</td>
</tr>
</tbody>
</table>

Suppose we have only four green tiles, and we have named them using one based numbering, that is starting the lower left tile at 1:1. The above illustration shows those four image tiles along with an empty matrix of where other tiles would be placed by name if they existed (and, in the case of green tiles, if we were to use zero based numbering).

Now, suppose we have created a single, new intermediate level image (shown in blue) that covers a region four times larger than a single green tile. If we name this image 1:1, see where it appears in the positioning matrix: it is not aligned to the lower left of the four green tiles but instead is aligned to the 2:2 green tile.
This is exactly what one would expect from the modulo arithmetic of the matter since $1$ divided by $2$ is zero using integer arithmetic. The $1:1$ green tile is covered by what would be the $0:0$ blue tile as can be seen from the illustration above.

So what would an actual image library consisting of many high resolution green images and four lower resolution blue images actually look like?

If we recall the situation shown above, we see that the four lower resolution tiles do not fully cover all of the green tiles.
If we open the image library and zoom far enough into the image library all we will see will be the green tiles. When we are zoomed far into the image Manifold will use only the highest resolution tiles.

As we zoom farther out, at some zoom level Manifold will begin using the blue image tiles since their lower resolution is a better match to the zoom level. In regions where there are no intermediate levels Manifold will have to interpolate on the fly from the higher resolution, green images.

Let's consider a photographic example using images of San Francisco Bay. We will use images that are all 200 x 200 pixels in size. Because they are identically the same size we can use the *Arrange images using filenames* option and manually specify an intermediate level image to use.
The overall image library will use nine images. Four of those images are seen above. These four images will occupy the lower left hand region of the image library.

We will also use one intermediate level image, seen above, which is also 200 x 200 pixels in size.
The intermediate image will cover the four higher resolution images in the lower left corner using the naming scheme shown above.

We will use the same mask as before so that the intermediate level image is called L1_00.tif and the nine higher resolution images have names such as L0_00.tif, L0_01.tif, L0_10.tif, L0_11.tif and so on up to L0_22.tif

When we open the image library window it looks like the illustration above, which has had an index drawing for the image overlaid on the image in a map window.

The index drawing has been formatted with transparent area background so the image can be seen. The area borders of the index drawing have been thematically formatted according to the value of the Tile Level field and labels have been automatically created using the X and Y columns from the index drawing. [The astute reader will immediately see that the illustrations for this topic have all been created in Manifold by using appropriately formatted index drawings.]

In the above illustration the intermediate level drawing does not cover all of the higher resolution tiles. It only covers the region within the red border. The regions outside of the red border are being interpolated by Manifold directly from the higher resolution images to create a seamless image mosaic regardless of which part comes from a higher resolution, lower level image and which part comes from a lower resolution, intermediate level image.

In fact, Manifold does such a good job at this that without the red border seen above it would be very difficult to tell by eye where the intermediate level image ends and where the higher resolution, non-overlapped images begin. Even if we carefully zoom in and out to be sure that we are seeing both the intermediate image and the high resolution images (just on the edge of the zoom value that would stop using the intermediate image), it is difficult to tell where one ends and the others begin.
It's easiest to see if we have a magnifying glass and can take a closer look at the actual pixels being displayed on the computer monitor. Seen above is a magnified view of a captured screen shot that shows the border between the intermediate image (lower part of the illustration) and the higher resolution images (upper part of the illustration). Note that the pixels in the lower image are "fatter" because they have to be inflated to cover four times the region each pixel in the higher resolution image can cover.

The above illustration shows one of two artificial effects at play in the illustrations in this documentation that are not typically found in image libraries in the real world:

- Usually when intermediate images are used they have been automatically calculated so that they cover any higher level, more detailed images. There is never an opportunity in such cases to see both an intermediate image and a higher resolution image together to make a comparison so we never see a difference in pixel size as seen in the image above. In such cases Manifold seamlessly switches between using different levels as appropriate for the zoom level displayed.

- To keep the example illustrations small enough to fit conveniently into this documentation, the intermediate image used above is only twice the X and Y extent of the higher resolution images. That makes the resolution difference relatively small so that the eye doesn't notice it as a great change even when the two resolutions are seen together. A more normal case would be to make a jump to four times the size so that each intermediate image covers sixteen higher resolution tiles.

**Tech Tip on Performance**

Image libraries are designed to work with large sets of small files, the kind that one gets when browsing an online mapping site such as MapQuest or Virtual Earth. Image libraries can be used with small sets of large files but performance will be significantly worse.

If we have ten large image files, it is better to use them directly without joining them into an image library. This can be done by showing the images together in a map with each image a different layer.

Alternately, if a single image is desired import the images into Manifold, uncompress them if they are in a compressed format, use Copy and Paste to create a single large image and then save the result as a single ECW or JP2 file. This process is tedious and very slow for large images but once the single large image has been created and saved as a ECW or JP2 file thereafter it will import and display with great speed.

The fastest way to store images is to use a spatial DBMS. Manifold can save images that are even tens of gigabytes into a spatial DBMS and then pan/zoom the image almost instantly.

**See Also**

**Compressed Images**
Image - Create Index Drawing
Intermediate Levels and Pyramids
Export Image - ECW / JPEG2000
Linked Images
Images and Channels

Images are often referred to as RGB images that use **channels**. Channels are a very simple idea that is easy to use.

**RGB Explained**

Much of how images are stored and manipulated in computers derives from the television technology that was first used to create computer graphics displays.

If we took a look at our computer monitor’s display with a magnifying lens we would see that it consists of a very large number of triplets of red, green and blue dots. Depending on the monitor manufacturer they may be round dots or small squares or other shapes but there always will be triplets of red, green and blue elements. In most monitors the dots are quite large and easy to see even with a weak lens.

Images on the computer display are formed when the monitor precisely varies the brightness of the red, green and blue elements in each triplet. Because the dots are close together the human eye will fuse the three red, green and blue dots of varying brightness into a single dot that appears to be the color combination of the three levels of red, green and blue color. For those of us who missed art class in school, all colors perceived by humans can be formed by the right brightness combination of red, green and blue color. Amazing, but true!

From here on in we’ll refer to the Red, Green and Blue elements of images as "R", "G" and "B".

Images on computers are usually nothing more than a series of number triplets where each triplet of three numbers is intended to control a single triplet of R, G and B dots on the monitor. Each triplet of numbers is a single pixel. A number triplet such as 67, 228, 180 means to turn the R element up to 67 brightness, the G element to 228 brightness and the B element to 180 brightness. The result will be seen by humans as a pretty shade of green-blue color. The RGB numbers are called **color values**.

Image files are arranged so that there is a series of pixels in rows that correspond to the rows of tiny triplet dots on the monitor. When an image is displayed on the monitor so that each pixel triplet in the image file controls one color triplet on the monitor we say that the image is "natural size" or "100% zoom".

In most modern software packages we take for granted the ability to zoom in or zoom out of an image to see it smaller or larger than natural size. If we zoom in, our graphics software will take the RGB values in a pixel that were originally intended to control just one color triplet and will use that same value across and down as many triplets as is necessary to make the image bigger on the monitor. If we zoom out so that the image is much smaller than intended, then there will be more RGB values than there are color triplets to control. In that case our graphics software will average out the RGB pixel values to figure out what average value should be used to drive the color triplet for that spot in the image. Manifold System performs these functions automatically.

**Channels**
How many numbers are used to specify the color of each pixel is the number of **channels** each pixel has. In RGB as described above, an image has three numbers for each pixel that directly correspond to the three R, G and B elements in the computer display. Such RGB images have three channels.

When we consider that an image that is 1000 pixels wide by 1000 pixels high contains a million pixels overall, if we have three numbers for each pixel (one number each to control the R, G and B dots) that can add up to very many bytes of data. If each number is just one byte, then a one million-pixel image will take three megabytes of space. Not surprisingly, there have been many clever software schemes invented to reduce the amount of space required for an image.

A quick and dirty scheme is to use just one number for all three RGB dots. If the brightness of all three dots is at the same level, we perceive no color but just different shades of gray. Using one number per pixel in a million-pixel image reduces the size to only one megabyte, but at the price of seeing the image in shades of gray as a **monochrome** ("black and white") or **grayscale** image.

How many numbers we have per pixel is the number of channels that image has. A monochrome image that has one number per pixel has one channel. A more typical image that has three (R, G, B) numbers per pixel has three channels. Such images are called **RGB** images. Although images with three channels are most common because of the universal use of RGB formats in color displays, there is no reason why an image could not contain more numbers per pixel for a greater number of channels than three.

For example, images used in sophisticated graphics editing use four channels: three channels for RGB plus an extra **alpha** or "a" channel. The alpha number says how transparent that particular pixel is supposed to be when it is combined in layers with other images. Such images are called **RGBa** images. Images coded with four, RGBa, channels can have regions of the image that are semi-transparent, so items in lower layers can be seen to a greater or lesser degree. Ordinary RGB images can have invisible pixels but this is a simple ON/OFF effect for each pixel. RGBa images can have a different percent transparency for each individual pixel in the image. This is called **pixel transparency**.

Another type of one-channel image uses a single number per pixel to specify which color in a palette of 256 colors is to be used for that pixel. See the Palettes topic for an explanation of what palettes are and how they are used. For the purposes of channels, it is important to note that images that use palette images are one channel images. This one channel is then expanded into three numbers per pixel via the palette.

Images collected by satellite sensors can contain hundreds of channels per pixel, with each channel being a number that represents some special quality collected by that sensor. Channels might code the height of that pixel, the temperature as seen from space or the reflectance as seen in various spectral bands invisible to human eyes. Since computer monitors only have three, RGB elements for each pixel, showing such multi-channel images requires software that maps the many channels into only three for display purposes. Such multi-channel images are often displayed using "false color" techniques that map one or more channel values into a specific range of RGB values.

There are two ways of working with multi-channel images such as multi-spectral satellite raster data images. First, such images may be stored as **.ecw** images and imported into Manifold as multi-channel compressed images. Channels from the compressed image may then be assigned to R, G, B and Alpha channels using the View - Display Options dialog. An alternate technique is to import multi-spectral images from non-.ecw formats into Manifold as multiple grayscale images where each image represents one channel. The grayscale images may then be combined three at a time to create false color RGB images or analyzed with scripts that can work with multiple images (in this case representing multiple channels) at once.

**Channel Controls**
When working with photographic images it is often very convenient to manipulate just one channel at a time. For example, if a photograph was taken outdoors without a blue sky filter the overall image may appear too blue. In that case, it would be very convenient just to see the blue channel and to turn down the "blue" intensity overall to get a more natural-looking image.

When used with images, the Layers pane (pop open from the View - Panes menu) will automatically display all the channels in whatever image has been opened. Click the checkboxes to turn channels off and on. Any editing effects will be applied to only those channels that are checked on and when painting into the image only those channels which are enabled will get painted.

The layers pane will show one channel for grayscale and palette images, three channels for RGB images and four channels for RGBA images. The layers pane also provides checkboxes to draw a background as well as a border around the image. See the Layers pane topic for more information on the border and background checkboxes.

Using all three channels shows the image in natural color.

Using no channels results in no visible image.
Checking just the Red channel shows the image using only the red dots on our monitor.

Checking Green only uses only the green dots. Note that the region of blue sky has darker red and green tones because it is predominately blue in appearance. The white clouds are bright in all three channels because simultaneous brightness in R, G, and B results in white.
Checking Blue uses only the blue dots. If we have a photograph shot outdoor with a sky filter correction it may end up too blue. To provide a more natural appearance, we might reduce the brightness of all blue tones to make the photograph less blue.

Using two channels together results in "combined" colors as used in the printing industry. For example, Blue and Red together are Magenta.

Blue and Green together are Cyan.
Green and Red together are Yellow (surprisingly!). Keep in mind when mixing RGB channels we are mixing light, not paint. Mixing green and red paint together results in mud, but mixing red and green light together results in yellow.

Commands use Visible Channels

All Manifold image commands work only on those channels that are enabled. For example, changing Brightness in an image with the Red and Green channels unchecked and only the Blue channel visible will affect the brightness of the Blue color value numbers only.

To make an image more or less yellow, check the Green and Red channels ON and the Blue channel OFF and increase or decrease brightness.

Likewise, painting into an image paints color values only into those channels that are checked ON. See the Painting within Channels topic for details.

Saving Channels as Separate Images

At any time we can save the separate channels of an RGB or RGBa image as different grayscale images. To do so, we use the Edit - Save Mask/Channel command with the channel to be saved specified in the dialog's Save box.

See the Masks topic for a discussion on how Edit - Save Mask/Channel can be used to separate an image into different images for each channel.

See Also

Painting within Channels
Separating Images by Channels
Combining Channels into Images
Colors as Hue, Saturation and Brightness

Describing colors using hue, saturation and brightness (also at times referred to as hue, saturation and lightness, or HSL) is a convenient way to organize differences in colors as perceived by humans. Even though color images on computer monitors are made up of varying amounts of Red, Green and Blue phosphor dots, it is at times more conceptually appropriate to discuss colors as made up of hue, saturation and brightness than as varying triplets of RGB numbers. This is because human perception sees colors in these ways and not as triplets of numbers.

If we imagine the three primary colors red, green and blue placed equally apart on a color wheel, all the other colors of the spectrum can be created by mixes between any two of the primary colors. For example, the printer's colors known as Magenta, Yellow, and Cyan are mid-way between Red and Blue, Red and Green and Blue and Green respectively.

This diagram is called the color wheel, and any particular spot on the wheel from 0 to 360 degrees is referred to as a hue, which specifies the specific tone of color. "Hue" differs slightly from "color" because a color can have saturation or brightness as well as a hue.

Saturation is the intensity of a hue from gray tone (no saturation) to pure, vivid color (high saturation).

Brightness is the relative lightness or darkness of a particular color, from black (no brightness) to white (full brightness). Brightness is also called Lightness in some contexts, in particular in SQL queries. See the Raster Extensions topic.

The illustration above shows the difference between saturation and brightness. We first pick a blue hue from the color wheel (stretched out into a line to make a prettier illustration). We can then reduce the saturation so that the blue hue becomes more and more blue gray. With zero saturation it is gray. We pick a less saturated blue tone and then turn the brightness up and down on that tone.
Note that increasing the brightness is not the same as decreasing saturation. Decreasing saturation turns the colors into gray shades. Increasing brightness turns a blue into a lighter sky blue but without making it gray.

Taking an almost totally desaturated blue, so desaturated it is almost perfectly gray, and increasing the brightness will result in what appear to be lighter shades of gray. However, if there is any "blue" at all in the gray by increasing saturation we can achieve a bright blue again.

This is why increasing saturation to a great degree in ordinary photographs will at times bring out unexpected, bright, vivid, posterized color effects. What has happened in such cases is that some "gray" pixels were not truly neutral grays but rather were desaturated hues. Increasing saturation to 100% saturation brings out the hidden hues.

**Numeric Definitions**

Hue, saturation and lightness (brightness) are computed from raw R, G and B channel values scaled into a range of 0 to 255 according to the following formulae:

- \( H \) (hue) = 0, if \( R = G = B \), otherwise a value between 0 and 255, with the range of 0 to 255 being split into three strips for G to B, B to R and R to G gradients.

- \( L \) (lightness) = \((M + m) / 2\), where \( M \) is \( \max(R, G, B) \) and \( m \) is \( \min(R, G, B) \).

- \( S \) (saturation) = 0, if \( R = G = B \), otherwise \( 255 \times (M - m) / (M + m) \), if \( L < 128 \), otherwise \( 255 \times (M - m) / (511 - (M + m)) \).

**Luma**, for historical reasons referred to as **intensity** is computed according to the following formula:

- \( I \) (luma) = \( 0.3 \times R + 0.59 \times G + 0.11 \times B \).
Invisible Pixels

Any image except a compressed image in Manifold can have invisible pixels, which do not appear in the image. They are simply placeholders for empty regions of the image. When images containing invisible pixels are stacked above one another in a map, items in layers below the invisible pixels will be visible.

Invisible pixels are a simple, low-overhead way of creating transparent regions within images. They are available within any Manifold image type. Using invisible pixels does not increase the storage size required for the image. The only disadvantage of the invisible/visible pixel facility is that pixels are either completely opaque or they are completely invisible.

If partial transparency is required, two other transparency methods may be used with images:

- **Layer Transparency** allows changing an entire image layer's opacity on a percent basis. This works with any image layer in a map. All the pixels in the layer will have the same percent opacity applied. This is a good effect for creating "see through" images for purposes such as aligning images to each other within maps or for creating faint images overlaid on maps to provide guidance. Layer transparency works with any image layer in Manifold and does not add to the storage size of the project.

- **RGBa Pixel Transparency** allows setting a different percent transparency for each individual pixel. This allows sophisticated gradients in transparency for the creating of drop shadows, composite images and feathering images. Pixel transparency is normally applied using editing tools such as partial erasers to partially erase various regions or by using gradients as masks for the alpha channel. Pixel transparency is available only within RGBa images, which are larger by one channel than they would be as RGB images.

Selections and other operations can occur with regions of invisible pixels; however, unless we paint color into them using some method such as the paint bucket they will not participate in image operations. For example, adjusting hue, saturation or lightness has no effect on invisible pixels. Painting color into an invisible pixel makes it a regular pixel again.

**To Create a Region of Invisible pixels**

1. Select the desired region.
2. Choose **Edit - Delete** from the menu or press the **Delete** key on the keyboard: The pixels will disappear from the image.

To restore invisible pixels into ordinary pixels, paint the region with any image drawing tool.

**Example**

Suppose we view the above two images in a map, with the bronze image above the schloss image. ["Schloss" is German for "castle" or "chateau". The image shows the famous Schloss Neuschwannstein in Bavaria.]

838
We wouldn't be able to see those parts of the castle image covered by the bronze image. We can look at the bronze image in an image window as well as in the map window.

The image window background has been turned off so we can see which parts of the bronze image become invisible. Manifold uses a checkerboard pattern to show regions where there is neither background nor any visible pixels. This makes it easy to see regions that are less opaque.
To create invisible pixels in the sky region, we simply select the pixels in the sky region and then *Delete* them.

If desired, we could have selected them in the map window and deleted them there.
When we delete the pixels they disappear. They will immediately disappear in the map window as well.

This makes that region of the bronze image completely invisible and allows more of the castle image to become visible.
Any operations applied to the image will not involve the invisible pixels. It is as if those regions of the image are not even there. For example, we can increase the saturation of the bronze image.

We can do this either in the image window or the map window without affecting the schloss image. Any operations we apply will affect only the visible pixels.

It's often the case that we would like to render invisible all pixels except those selected.
Suppose we've selected the region of pixels in South America and we would now like to make all other pixels invisible. We use **Edit - Select Inverse** to invert the selection.

Now we can use **Edit - Delete** to delete the selected pixels and make them invisible. The image is shown with the **Border** turned on in the Layers pane so we can see the size of the image.

**RGBA Pixel Transparency**

We can use RGBA pixel transparency in images that have transparent pixels. If the bronze image is an RGBA image, we can vary the transparency of each pixel independently.
For example, if we load the above grayscale gradient as a Mask for the alpha channel in the bronze image we are telling Manifold to vary the transparency guided by the grayscale value of the gradient.

Since the grayscale value increases from black to white this will increase transparency in a continuous gradient from zero transparency at the top of the image to full transparency at the bottom of the bronze image.
It's easy to create complex images mixing many layers of other images with suitable use of invisible pixels, RGBA pixel transparency and transparent layers. We can layer images of different sizes and move about the different layers to achieve the effect desired. For example, if we wanted to add pigeons flying about the statue we could cut pigeon images out of a picture with pigeons and then paste them into layers above the bronze layer, with one pigeon per layer if desired. We could then move the pigeons about by moving their layer, warp the pigeons with perspective effects to make them appear as seen from different angles and perhaps reduce a few in size and desaturate and lighten them to make it seem they are in the distant background.

Just for fun, let's Colorize the schloss image into a light sepia tone to allow the colors of the saturated bronze image to better stand out. We have also taken the liberty of selecting and then deleting the remnant pixels for the upper part of the clouds in the bronze image. The improvement in the composite effect can be seen by comparison with the illustrations below, where the screen shots were made before trimming the clouds and thus a fine outline of cloud layer can be seen in front of the castle.

**Invisible Pixels vs. RGBA Pixel Transparency**

RGBA pixel transparency is a different mechanism than invisible pixels. Key differences include:

- Invisible pixels are an "all or nothing" effect: either the pixel is fully visible or it is invisible. Alpha transparency allows a different transparency setting for each individual pixel.
• Deleting a pixel to make it an invisible pixel is a permanent deletion. The pixel with its original color values cannot be retrieved. Alpha transparency, in contrast, can be changed at any time to restore the original appearance of pixels.

• Using invisible pixels does not add to the size of the image. Alpha transparency works only with RGBA images and requires a fourth channel for each pixel whether transparency is used or not.

• Invisible pixels do not take any longer to process than regular pixels. RGBA images take longer to display because transparency for each individual pixel must be reckoned.

**Use with Projections and Drawings**

Invisible pixels work fine when combined in maps with drawing layers or with projections.

![Image of world map with sea areas removed](image)

A frequent use is to remove sea areas from photographic images so that graticule lines or other drawing elements in layers below the image can be seen.

Using **Select Touch** we selected the pixels in sea regions and then deleted them to create the above image in Mollweide projection. See the Invisible Pixels and Selection topic for information on selecting invisible pixels and on using the Invisible Pixels selection in the **Selections** pane.

**Cutting, Copying and Pasting and Invisible Pixels**

Some commands will automatically create regions of invisible pixels.

**Cut**

Cutting a selection out of an image leaves a region of invisible pixels.

**Copy / Paste**

Copying a selection takes only the selected pixels. When pasting from the clipboard into a new image layer, the "empty"
region in the rectangular image will consist of invisible pixels.

**Painting over Invisible Pixels**

We can create "real" pixels in regions of invisible pixels by simply painting into those regions using any painting or drawing tool that works with images.

Suppose we delete the sky pixels in our sample *bronze* image to create a region of invisible pixels.

We can see the preview of this in the Selections pane using the Invisible Pixels saved selection.
We can paint onto the region of invisible pixels however we like. In the illustration above we've added dots and other figures in green and gold.

Once we paint onto a region those pixels are no longer invisible. The Invisible Pixels saved selection, for example, no longer includes them.

**Invisible Pixels in Masks**

Any invisible pixels in masks will not participate in whatever the mask is asked to do. If the mask is loaded as a selection, regions of invisible pixels will not be selected. If the mask is loaded as a channel, any parts of the mask containing invisible pixels will not alter the target image.

**Saving Invisible Pixels as a Separate Image or Mask**

At any time we can save the region of invisible pixels as an image within an image. Such images are called masks and may be used at any time in the future for subsequent selection or channel operations. To save the region of invisible pixels as a mask, we use the Edit - Save Mask/Channel command with the Invisible pixels system saved selection specified in the dialog's Save box.

At any time we can use Edit - Load Mask/Channel to load a mask that specifies a region of invisible pixels. For example, we may have previously saved some selection as a mask and now we would like to use that selection to create a region of invisible pixels in the shape of that selection. We can use Edit - Load Mask/Channel with Load set to Invisible pixels and the region of invisible pixels will be set by the mask being loaded. Note that any previously invisible pixels will remain invisible since making a pixel invisible is always a "one-way" operation that deletes the pixel.

See the Masks topic for a discussion on how Edit - Save Mask/Channel can be used to save different regions for use as selections. It's easy and fun.
See the Invisible Pixels and Selection topic for information on selecting invisible pixels and on using the Invisible Pixels selection in the **Selections** pane.

**Invisible Pixels are not Deleted**

Note that when "deleting" pixels the pixels are not really deleted - they are simply made transparent. The only way to delete pixels in an image is to crop the image, in which case the cropped pixels are genuinely deleted.

Consider the example **SanFran** image on the Manifold CD that shows a LandSat image of the San Francisco Bay area. Note that it has regions of black pixels because the LandSat image is tilted. The black pixels are necessary because all images are rectangular (made up of a rectangular array of X times Y pixels).
We can select the black pixels and delete them.

However, deleting the pixels simply makes them invisible. The image still has the same original extent as it did before. The "deleted" pixels are still there as invisible pixels.

We can see that by turning on the Selections pane for this image and previewing the built-in [Invisible Pixels] selection in blue preview color. Because images must be rectangular there is no way to cut off triangular portions of them.
If we want to reduce the size of the image so that it contains no invisible pixels we must make a rectangular selection and then crop the image to that selection.

This will genuinely delete the cropped pixels, albeit at the cost of cropping the image down to a rectangular region that covers only visible pixels.

**See Also**

- Use the Layers pane with images to click ON a background color layer or to show a border around an image.
- About the Sample Images provides notes on the *schloss* and *bronze* sample images.
Invisible Pixels and Selection

By default, selection methods do not select regions of invisible pixels. The one exception is **Touch Selection**, which will select invisible pixels when they are touched.

Suppose we delete part of our standard *bronze* image to create an image with invisible pictures. We've also drawn a blue bar across parts of the image to divide it into upper and lower halves. We have turned on the **Border** in the Layers pane so we can see the actual size of the image.

If we click near the lower right corner with **Touch Select**, all the invisible pixels contiguous to the spot touched will be selected. Note that the contiguous selection "flows" only through the actual extent of the image. Therefore, it did not select "around" the blue band or the lower leg of the monument.
If we **SHIFT-click** with **Touch Select** we can select all of the invisible pixels in the image no matter where they are located.

**Other Selection Commands**

All other selection commands will ignore invisible pixels by default.

**Use the SHIFT key to Select Invisible Pixels**

**Select Touch** will select contiguous invisible pixels when the mouse is clicked on a region of invisible pixels. All other mouse selection commands, such as **Select Box**, will select only visible pixels within the mouse selection region. To select invisible pixels as well, hold down the **SHIFT** key while making the mouse selection.

A **SHIFT - Select Touch** command will select all invisible pixels in the image, whether contiguous or not.

A **Select Circle** selection does not select the invisible pixels. It only selects visible pixels within the selection circle.
Using **SHIFT - Select Circle** selects both invisible and visible pixels within the selection circle.

**Invisible Pixels Saved Selection**

Precisely because invisible pixels are not seen there are times when we might like some help in seeing them.

Invisible pixels are very frequently used when composing sophisticated images (that is, maps) that consist of stacks of many constituent image components. When working with many different image components that contain transparent regions we might forget which parts are transparent and which regions are simply white pixels in images or “white space” in drawings seen through the layer stack.

When working in an image window we can see which regions are invisible pixels by working with the background color turned off to allow the checkerboard pattern to show through. However, in the case of RGBA images it is possible to set RGBA pixel transparency to full transparency to make the pixels entirely disappear. The effect is the same, but these are not invisible pixels. They are visible pixels with their alpha values set to full transparency for each individual pixel and so the checkerboard pattern will be seen through such pixels as well.

In addition, we often would like to use the region of invisible pixels in selection combinations with selections made in other parts of the image. To support this usage and to make it easy to unambiguously see the region of invisible pixels Manifold devotes one of the seven saved selections in images to a system-provided saved selection that always shows the region of invisible pixels. This saved selection will appear in the Selections pane if any invisible pixels occur in the image.

Suppose we have an image that may contain invisible pixels.
To see the region of invisible pixels, simply click on this saved selection to highlight it and press in the **Preview** button in the Selections pane toolbar.

Any invisible pixels in the image will appear in blue preview color.

**Example**

This example shows use both of the invisible pixels saved selection as well as the use of the SHIFT key to select invisible pixels in addition to visible pixels.

We will use the `globe.bmp` sample image with contrast increased (to make touch selection easy in water areas) and relief added for a pretty appearance.
Using **SHIFT - Touch Select** we touch regions with water to select them throughout the image.

Choose **Edit - Delete** and the pixels disappear. (We have the white **background** color turned on in the Layers pane so the background appears white).

However, the invisible pixels are still there as placeholders for the region they occupy. We can click on a region of invisible pixels with **Touch Select** and they will be selected. They are simply invisible. Once selected we can pour color into them with the paint bucket to make them visible again.
Selections made over invisible pixels do not normally select them. Here we have made a Select Circle selection that ignores the invisible pixels.

To select invisible pixels, we hold the SHIFT key down while making the selection. For example, a SHIFT - Select Circle command will select the invisible pixels within the circle as well as the visible pixels.

It’s often the case that we would like to see the region of invisible pixels or use it with selection combinations. Clicking the invisible pixels saved selection in the View - Selections pane will show it to us in blue preview color if the Preview button is pushed in.
We can use the invisible pixels selection like any other saved selection. For example, we can use **Subtract from Selection** to subtract the region of selected pixels from the circle selection.

If we push out the **Preview** button to remove preview color we can see that we have just made a circular selection within Africa that is "clipped" at the coastline. This is a somewhat contrived example since we could have done the same thing just by selecting the circle without holding down the SHIFT key as we did in an earlier step. However, there will be many times when it will be convenient to use the invisible pixels saved selection in **Replace, Add, Subtract, Invert** or **Intersect** commands together with the existing selection.
RGBa Pixel Transparency

RGBa images can have a different transparency level for each individual pixel in the image. This is achieved by adding an alpha channel to the three basic R, G and B channels. The effect is also known as alpha transparency. Compressed images can also utilize alpha transparency by choosing an image channel to be used as the alpha channel via the View - Display Options dialog.

The appearance of each pixel in RGBa images is defined by four numbers: the three R, G and B numbers that specify the color and an alpha channel number that specifies the transparency of that particular pixel in a range from 0 (fully opaque) to 255 (completely transparent). The ability to work with RGBa images is one of the key capabilities that distinguishes a professional-class image editor like Adobe Photoshop or Manifold System from consumer-grade graphics editors.

If you have not yet done so, please read the Images and Channels topic before proceeding with this topic.

To use RGBa Pixel Transparency:

1. Convert the image to RGBa using Image - Convert to
2a. Paint any gray color into the alpha channel of the image using any paint tool; or
2b. Load a grayscale mask for the alpha channel using Edit - Load Mask/Channel

RGBa pixel transparency works only with RGBa images. If the image is converted to any other type (RGB, grayscale or palette) the partial transparency effects will be lost.

Alpha transparency does not change the R, G, and B channels that define the color value of the pixel. When we appear to “erase” pixels using alpha transparency the pixels and their original color values are still there. They are simply transparent because their alpha channel values have been set to high amounts. If we change our minds about any transparency effects applied we can restore the original appearance of the pixels by converting the image to an RGB image.

See the Painting within Channels topic for examples of painting gray color into the alpha channel to control alpha transparency.

Example

Suppose we have an image of Europe using mostly blue colors.
We can make a copy of the blue Europe image and use Hue / Saturation to change the color range to red.

Using the partial eraser we can erase most of the image leaving only the central portion with a gradual transition from fully opaque pixels to fully transparent pixels.

If this image is layered in a map above the blue Europe image we get the effect seen above.

We can also use masks, such as the grayscale gradient above, to control alpha transparency.

When the mask above is loaded into the alpha channel of the red Europe image it causes zero transparency in black regions of the mask and complete transparency in white areas of the mask. The regions of the mask that transition from black to white cause a gradient of partial transparency in the red Europe image.
Seen in the map above the blue image the result is a gradual color transition from Northwest to Southeast.

There is no limit to the artistic and presentation effects made possible by alpha transparency. We can adjust the region of transparency using a wide variety of Manifold editing tools.

The illustrations above show the red Europe image as a separate image to make clear what is going on. Many times we will edit alpha transparency free-hand in maps using editing tools such as the partial eraser to see the effect in a many-layered map in real time.

We can create many special effects by making copies of an image and then adjusting the different copies in different ways (such as by Hue / Saturation or other commands), partially erasing pixels in the variations and then recombining them.

**Uses for Alpha Transparency**

- Creation of complex images using many layers.
- Graphics art effects. See the Invisible Pixels topic for an example.
- Drop shadows, halos and "stand outs". See the Gaussian Blur topic for an example.
- "Feathered" transitions from one image to another.
- Color range effects as shown above.
- Transitions between different effects, such as blur to non-blurred images.
- Fine feathering of images to eliminate anti-aliased pixels at the edges of "cut outs."
- To give the appearance that text or other images are painted onto a surface.
- To merge tiles of different images into one image without obvious borders.
To paste patches into images to hide defects or unwanted imagery without an obvious border line.

**Example**

Manifold System has such powerful image manipulation capabilities that it may also be used to create complex images by combining multiple images in layers as is done with professional quality graphics image editors such as Adobe PhotoShop.

The marketing promotion "splash" screen used with Manifold System Release 5.00 was created using over 40 layers of other images. Alpha pixel transparency and invisible pixels were used throughout the various layers so lower layers could combine to form the image. Looking at seventeen of the various layers we can see how the image was created:

The bottom three layers were screen shots from a Terrain window using different settings for sky and surfaces. The uppermost of these is the grid.

The next two layers are the floating city and its shadow, which are in separate layers so that the transparency and relative position of the shadow may be easily controlled. Once the shadow was in the right position relative to the city, the Move Together box for both was clicked for these two layers so that moving the city automatically moved the shadow as well. See the Gaussian Blur topic for details on creating drop shadows.

The next twelve layers consist of text or logotype images together with various drop shadows. By checking and unchecking the Move Together boxes as we click on any of the layers, we can move various layers together to compose the image as desired.
The actual production "map" for the splash screen included over 40 layers. The extra layers contained variations of the logotype, different visual elements such as different sky cloud patterns and other variations. By checking the layers ON and OFF and moving the various layers about it was easy to try out many different variations to end up with exactly the effect desired.

Each image in the layer stack was heavily manipulated using various effects, resized, partially erased or made transparent to "feather" its features visually into features from other layers, and so on. For example, the city originated in a scanned photograph snapped from the bell tower in St. Mark's square in Venice. It was cut out of the photograph and then different parts of the image had hue, contrast and color balance adjusted in different ways. To bring out detail in the trees the green areas were re-built from several layers using different settings for brightness, contrast, saturation, colorization and color balance.

See the Composing Complex Images in Layers topic for an additional example of how to compose complex images using Manifold.

**Saving the Alpha Channel as a Separate Image**

At any time we can save the separate channels of an RGB or RGBA image as different grayscale images. To do so, we use the Edit - Save Mask/Channel command with the channel to be saved specified in the dialog's Save box. For example, we can save the gradations of an alpha channel as a grayscale image by choosing Channel: alpha in the dialog's Save box. This is a very useful way of saving different transparency patterns. We can try out different amounts of alpha transparency as we like and always return to one pattern that we like the best.

See the Masks topic and the Separating Images by Channels topic for information on how Edit - Save Mask/Channel can be used to separate an image into different images for each channel.

**Invisible Pixels vs. RGBA Pixel Transparency**

Alpha transparency is a different mechanism than invisible pixels. Key differences include:

- Invisible pixels are an "all or nothing" effect: either the pixel is fully visible or it is invisible. Alpha transparency allows a different transparency setting for each individual pixel.
- Deleting a pixel to make it an invisible pixel is a permanent deletion. The pixel with its original color values cannot be retrieved. Alpha transparency, in contrast, can be changed at any time to restore the original appearance of pixels.
- Using invisible pixels does not add to the size of the image. Alpha transparency works only with RGBA images and requires a fourth channel for each pixel whether transparency is used or not.
- Invisible pixels do not take any longer to process than regular pixels. RGBA images take longer to display because transparency for each individual pixel must be reckoned.

**Performance Considerations**

Working with RGBA images is slightly slower than working with regular RGB images because of two factors:

- RGBA images are larger than RGB images.
- Adjusting transparency for each individual RGBA pixel takes longer than simply displaying an RGB pixel.

Most graphics art editing of images is done with images that are tens or hundreds of times smaller than images can be in GIS applications. For that reason using RGBA and alpha transparency is not normally a performance issue with graphics arts image editing in applications such as Adobe Photoshop.

Because GIS images such as raster data images can be well over a hundred megabytes we need to consider in a GIS environment whether it is worth it or not to convert images to RGBA and to use alpha transparency. Operations with such large images on desktop machines should consider every performance issue.

**Nomenclature**
Transparency and opacity are two terms that mean the same concept viewed from different directions. When something is completely opaque it is not at all transparent. When something is perfectly transparent it may be said to have zero percent opacity.

Which word is used depends on the discussion. When imagining layers stacked up above each other like transparent sheets it is conceptually clearer to use the word transparency. When discussing a specific percentage of light transmission to be applied via a slider bar in a dialog most applications use the word opacity.

The convention in the graphics arts editing software industry is to adjust layer transparency with controls that set a number from 0% to 100% opacity, so that an image with 100% will be fully opaque and not allow any view of an image underneath it. Manifold follows this convention. This convention persists in the graphics arts industry even though the technical implementation of transparency effects is done using an alpha channel within RGBA images where the higher the value of the alpha channel (from 0 to 255) the higher the transparency.

One therefore encounters the slight conceptual dissonance of increasing opacity with higher numbers (up to 100%) in dialogs and other user interfaces while the internal data sets use numbers (alpha channel values) in which opacity decreases with higher numbers. Since we rarely set alpha values by hand this is not so bad. Alpha values are normally set using various tools, such as erasers, or masks. In the case of masks, the darker the mask region the lower the alpha value is and thus the higher the opacity. From a casual conceptual view this is very acceptable because it leads to an effect where black regions of masks cause full opacity and white regions of masks cause full transparency. Since we are used to thinking of “white space” as being transparent this works well as a natural mnemonic for the effects of masks.

See Also

Use the Layers pane with images to turn on a background color layer or to show a border around an image.
Palettes

Palette images save image space by reducing the number of colors used in the image to 256 colors and then using one number per pixel (one channel) to specify the color for each pixel. Each color number corresponds to a color in a palette of 256 colors. Each color in the palette is a True Color RGB color out of a possible range of millions of colors.

Manifold uses the terms “palette images” and “palettes” because those are the terms used in Microsoft standard applications such as Microsoft Photo Editor and Microsoft Office. Non-Microsoft graphics packages and legacy GIS applications may use different terms. For example, palette images are known as Indexed Color mode images in Adobe PhotoShop and palettes are called color tables or color maps in some applications.

Palette images have become less important for graphics editing in modern times because the cost of disk drive space has plummeted. In an era when cheap disk drives deliver tens of gigabytes it is no longer as important as it used to be to reduce the space consumed by images. Preserving high visual quality is usually much more important with most images than saving space.

The exceptions are massive raster data images that originate in various technical programs, which may be tens or even hundreds of megabytes in size. Reducing the size of such images by converting them to palette images is often a wise trade-off between quality and size.

Another reason for use of palette images is that many images are published in formats that use palettes. Manifold supports fine control over palette images and the palettes they use to allow maximum exploitation of such images.

Another reason to use palette images is to reduce the size of images that will be published on Internet. Even with fast connections, Internet is so slow that it is very important to reduce the size of images used on web sites. Reducing the number of colors in an image allows web graphics formats such as .gif and .jpg the best possibilities of compressing the image to a small size.

Finally, an important raster data usage of palettes is to force pixels into a pre-defined set of color values. Such values can represent various classification schemes for the physical regions represented by the pixels or other data values.

About Palette Images

RGB images use three numbers per pixel to specify the colors in an image. Using three numbers per pixel allows about 16 million color combinations in most installations but it requires a lot of storage space for each image. As mentioned in the Images and Channels topic there have been many methods invented for reducing the size of RGB images.

One method noted uses one number per pixel by averaging brightness for all three pixels to provide a single, grayscale number for each pixel. This is very compact for storage, but it loses all non-grayscale colors in the image. Another popular method for reducing the size of RGB images is to use a palette. This method will preserve good color appearance in many cases while dramatically reducing the size of the image.

The palette method depends on the observation that most RGB images use only a small number of the 16 million possible colors available for use. There are two reasons why this is so. The first reason is that most images have far fewer than 16 million pixels in them. A 1024 x 768 image, for example, has less than a million pixels. Even if each pixel were a different color such an image would require much less than 16 million different colors to draw.

The second reason is that most colors used in computer images are so similar to each other that they are indistinguishable to the human eye. An example:
If we zoom far into our example bronze image near the head and chin of the monument we can see that the blue sky is composed of many blue pixels that are slightly different shades of blue. However, most of the blue shades are so similar to each other that it is very difficult to tell them apart. We could easily replace the hundreds of different, but virtually identical, shades of blue with two or three “blues” without significantly changing the appearance of the image.

That's exactly what the palette method does: to convert an RGB image into a palette image the system analyzes all RGB colors in the image and groups them into similar shades. The palette method then takes all the very similar colors in a group and replaces them with one color that is a close match to all of the colors in that group. For many images, it is possible to choose only 256 colors that are so reasonably close to the various similar colors out of all 16 million possible colors that the image appears the same as the original.

To see this in action, consider the above two images. Both images are zoomed in views of the bronze sample image showing the head and chin of the bronze monument. The image at left uses three RGB numbers per pixel so that each pixel can be any one of 16 million possible colors. Note how the blue sky is composed of many similar shades of blue, most of which are indistinguishable from each other.

The image on the right uses only 32 colors throughout the entire image. Each of these 32 colors is an RGB color chosen from the overall range of 16 million colors to be the best match possible to the group of similar colors it replaces. Only two shades of blue, for example, appear in the blue sky. We use 32 colors in this example because using 256 colors as is normally done with palette images results in such a close match to the original image that it is hard to tell the difference. Using 32 colors makes differences more apparent for the sake of an example.

The palette method saves the reduced number of colors in a palette (also known as a color table or color map) which is to be used for that image. Each color in the palette is listed as a three-number RGB color. The 32-color palette for the sample bronze image converted to 32 colors (part of which is shown above), for example, is:

Using only the above 32 colors results in an image that is remarkably close to the original.
The image on the left is the original, while the image on the right is the 32-color image.

**How Palettes Work**

Palette images in Manifold use 256 colors per image. This makes it possible to save images using only one byte per pixel, since each byte can represent 256 colors (bytes range in value from 0 to 255 for 256 positions in all). The first step in converting an RGB image is reducing it to 256 colors as described above. This requires picking out a palette that is a "best match" to the many colors in the RGB image and then assigning a color from the palette to each pixel in the image. Each pixel then is represented with one number that specifies which color in the palette is to be used for that pixel. All this is done automatically in one step by the Image - Convert To dialog when it converts an RGB image to a palette image.

*For programmers:* Palette images have one byte per pixel where the byte is an index into the palette. The palette is just a look-up table that says what RGB value is to be used for that pixel. That is why Adobe PhotoShop refers to such images as "Indexed Color" images.

Let's take a look at how the example 32-color bronze image might be saved as a palette image.

The pixels in the region enclosed in red in the illustration would be coded as one number per pixel:

<table>
<thead>
<tr>
<th>6</th>
<th>8</th>
<th>15</th>
<th>16</th>
<th>16</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>9</td>
<td>15</td>
<td>16</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>15</td>
<td>16</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>15</td>
<td>16</td>
<td>16</td>
<td>16</td>
</tr>
</tbody>
</table>

The number for each pixel says which color in the palette to use for that number. To figure out what color the number 16 in red shown above is supposed to be, we look at the palette used for this image:
All the colors used in this image are listed in the palette.

The look-up process is simple: the number 16 takes us to the sixteenth color in the palette, which is a shade of blue specified by the RGB triplet of numbers 91, 143, 207. These three numbers will be used to control the RGB pixel elements in the computer monitor for that specific pixel.

Note that the pixel to the right of the one shown in red has the value 18, which is a lighter shade of blue, the eighteenth color in the palette, two slots over from 16.

A Useful Analogy

If the above sounds confusing, perhaps a Liberal Arts analogy will help. Many people during their childhood have had exposure to "paint by numbers" kits. These kits provide white drawing boards that are covered with faint lines denoting regions that are to be filled with paint. Each region has a number in it. The numbers correspond to a collection of paint colors that are provided with the kit. Each color has a number. The child's task is to paint each region with the color paint corresponding to the number in that region. For example, all number 5 regions should be colored using paint from the number 5 container, which might be red. If done carefully the result will be a reasonable picture.

In classical artistic circles, the word "palette" means the board with colors on it that a painter holds in his or her hand. The painter chooses colors from the palette to load onto a brush that applies those colors to the canvas. Some painters in art history had a preference for using a particular selection of colors on their palettes and so the word "palette" in art history has also come to mean a particular collection of colors favored by a painter. One can speak of the distinctive palette of Cézanne, for example, or one might compare the sepia palette used by Van Gogh in his early work with the vivid palette he employed late in life in Arles.

In a paint-by-numbers kit the palette is the numbered set of paint colors. Palette images in computers are like the drawing boards in a paint-by-numbers kit where the number at each spot says which dab of paint to apply at that spot. It's obviously a lot more economical to code each spot with a single number than to write out "Ocher Brown" at one spot and "Goldenrod Yellow" at another.

The analogy is even more useful if we recall another childhood incident we may have experienced or heard about. Suppose in the coloring-in of our paint-by-numbers kit we get confused and think that the number 5 means to paint in a blue color and not the red color. The result will be that throughout the entire image we have switched the palette color for red with blue. In Manifold, such changes in palette colors are something that may be deliberately done with palette images in order to achieve desired color effects.

Images and their Palettes
Each image that is a palette image will have an associated palette that it uses. The project pane will show each palette underneath the image it serves.

Note that a logical outcome of the palette method for images is that for photographic or visual rendering the palette used with each particular image is specific to that image. The particular set of 256 colors that does the best job of approximating colors in one RGB image will not be the same as the 256 colors used to best approximate a different image.

For example, the sample *bronze* image is mostly greens and blues so the 32-color palette for this image shown above has many different shades of green and blue.

The sample image of *tokyo* airport, on the other hand, would use an entirely different palette that has many shades of near-black color when reduced to 32 colors.

**Changing Palettes**

Most graphics editors either do not show palettes that are used by palette images or they bury access to the palette deep within specialized menus. They do this because it is rare that one changes the palette in a photographic image that is only intended to provide a good visual appearance to the human eye. Images in a GIS environment are often used for other purposes besides providing a natural, photographic view of a scene. We will want to change palettes sometimes for such other purposes. Manifold System therefore provides the palettes used with images as separate components immediately at hand in the project pane.

We may wish to change palettes to achieve sophisticated image effects with visual images or to better represent data that is shown as a raster data image using a palette. Raster data images, for instance, are often created from multi-channel satellite images using “false” color to show data in the satellite scans. The colors used in a palette for such images might be arbitrary choices intended to show data in some range. For example, a palette image might show temperatures at particular locations using a palette where the colors range from blue to green to yellow to orange to red to show an increasing temperature range.

When working with such images in a GIS environment Manifold makes it easy to pop open the palette used in a palette window if changes to the palette colors are desired. For example, one might wish to change the colors in the temperature range palette so that the very coldest colors are all blues and the very hottest values are all reds with nothing but one shade of gray in between. To do this we would select all the colors between red and blue in the palette and change them to gray. All pixels in the image that were colored using those palette colors would instantly switch to gray. It's just like changing the paint containers in a paint-by-numbers kit.
Double click on a palette component to open it in a palette window. Make any changes by clicking into the color wells for individual colors and changing the colors. All standard Manifold selection methods will work when selecting colors in the palette window. See the Editing Palettes topic for an example.

Any changes made to a palette will immediately be applied to the image that uses that palette.

**Palette Sizes and Image Sizes**

Using a palette involves adding about 1000 bytes to the image size to save the palette lookup table (256 palette positions having three numbers each adds up to 768 bytes). For images that are thousands or millions of bytes in RGB form that is a big economy because investing 1000 bytes into a palette cuts the image size down by two-thirds at least. Some image formats using palettes can compress the image size even further if the image is exported into those formats.

Manifold palettes are always 256 color palettes. Images may be adjusted so they use fewer than 256 colors out of the palette. The main reason for using fewer than 256 colors is to simplify data management when using palettes with false color raster data images. It’s easier to assign different colors if there are only 8 colors, say, in our range of temperatures for a raster data image than if we need to assign 256 colors for different temperatures.

Although there is no savings in image space used within the Manifold project if an image uses fewer than 256 colors, using fewer colors may result in space savings if the image is exported to certain formats. Some graphics formats, such as GIF, will automatically compress the image to a smaller size if the number of color changes is fewer. Keeping the size of the image as small as possible is important when creating images for use on web sites, where every byte of the image must be flung down a potentially very slow Internet connection.

Using as few colors as possible will give the web image format used the best chance of compressing the picture when it is exported for use as a web image. Another way of keeping images as small as possible for web use is to use a standard palette, such as the standard web browser palette.

**Standard Palettes**

The use of palette images is so widespread that operating systems and web browsers are set up to exploit their use.

Although using a palette always provides big savings in space with large images, when images are very small the thousand bytes required for the palette becomes a significant percentage of the total size of the image. For example, one thousand extra bytes are a lot of bytes if the image itself is only a 32 x 32 icon (1024 bytes). In an operating system where numerous small images may occur it would be very inefficient if each small image had to carry its own palette around with it. To avoid this inefficiency Microsoft Windows includes a standard palette that is built into the operating system. Images that use the standard palette don't have to include a copy of their palette. They simply use the operating system palette.

Many web pages include numerous small images used as buttons or other graphical elements. If each used a unique palette, the size of such images could effectively double. To economize on download bandwidth all Internet web browsers include the same, standard palette for use by web pages. Using the standard web palette for small images destined for a web site will often reduce their size. Unfortunately, since the first web browsers were designed by people who were oblivious to the interests of the general computing public a different palette is used in web browsers than that made standard by Windows.

Both the web palette and the Windows palette incorporate a selection of colors that will work reasonably well for most images. Neither will work quite as well as a palette adapted specifically to a particular image but both will often work well enough for operating system or web use. Manifold's Image - Convert To dialog for creating palette images provides a list box that allows choice of a standard palette if one is desired. If a standard palette is selected the image will automatically be converted using only those colors that are in the palette selected. Standard palettes include both Windows and web palettes. For those with a sense of humor there is even an option to use the Mac system palette.

**Importing / Exporting Palette Images**

Some graphics image formats use palettes and some do not. When importing an image from a format that uses palettes Manifold will automatically import and create a palette component for the image.
When exporting to a format that uses palettes Manifold will combine the image and palette components in the correct way for storage within that format. The palette used with each image is embedded within the image file for all standard graphics formats that use palettes.

Creating a Palette Image

Palette images and their palettes are created automatically when importing from a graphics format that uses palettes. Palette images may also be created by converting from grayscale, RGB or RGBA images.

Use the Image - Convert To dialog to convert an image into a palette image. A choice of conversion methods and standard palettes will appear in the Palettes list box.

When to Use and Not to Use Palettes

It is important to understand that converting an RGB image into a palette is a permanent change in the data of the image. An image that formerly could have consisted of as many different colors as there are pixels in the image (up to a total of about 16 million different colors) will be simplified into using only 256 colors.

Reducing an image to only 256 colors will often result in negligible change in visual appearance. However, certain finely detailed images will lose vividness and may get a granular appearance in regions of smooth color gradient changes. If images must retain as much visual quality as possible it is best not to convert them to palette images. Images destined for enlargement or professional printing should never be converted.

Not all image commands will work in palette images. For example, Hue / Saturation depends on having three RGB channels for fine control of shades and will not work in one channel images such as palette images.

When creating complex images using RGBA transparency and other sophisticated effects all editing should be done using RGB or RGBA images. Only after the final image is ready should one consider converting it to a palette image and then only if it is destined for publication on the web. In an era of very inexpensive disk drives that deliver many tens of gigabytes of storage it is more important to keep visual quality high in photographic images than it is to save disk space.

Many raster data images in the mapping world originated as raster data sets in circumstances where each pixel's data had some scientific importance. Despite their technical origin such raster images often end up being used in a purely visual way where the specific data attached to each pixel is no longer important. Such images in their original raster data form can be enormous, with sizes in the hundreds of megabytes. It's a good idea to convert such images to palette images to save space and to increase system performance.

However, if the multi-channel data for each pixel continues to be important for some analytic purpose they should not be converted since doing so will destroy the data.

Notes

The above examples are written as if the "original" images use 16 million colors and the palette images are 32 colors. In fact, all images that appear in this Help file have been reduced to at most 256 colors due to limitations of the Windows Help system. Using 256 colors to represent the "16 million-color" images nonetheless results in such a good effect that we still had to use 32 color examples for palettes in order for the effects of conversion to be obvious.

The numbers in the palette example were made up by glancing at the image and estimating what the numbers should look like inside the image. They show the idea and relationships correctly but should not be taken as the exact numbers that will appear when the *bronze* image is converted into a 32-color palette image.
Images and Surfaces

Surfaces are data sets that may be visualized in 2D or 3D. They often provide terrain elevation data so that the heights of a given region may be seen in 3D. Although surfaces are imported from a variety of formats, once imported into a Manifold project they have many operations in common with images.

When a surface is opened in a surface window or appears as a layer in a map, the appearance of the surface is controlled by options in the View - Display Options dialog. By default, the surface appears as a grayscale image with color-coded by elevation.

Note that although surfaces appear in windows like images and share many commands that may also be applied to images, they are not images. The appearance of the data within the surface window is controlled by the display options, some of which (such as slope or aspect) may involve computation. In contrast, the appearance of images within an image window is "hard wired" by the color values of the pixels.

If desired, a surface may be converted into an image by copying the surface and pasting it as an image. In that case, the colors in the image pixels will be taken from the display options currently set for the surface. We might wish to convert a surface to an image to allow manipulation of the image for presentation purposes. Another way to make an image from a surface is to open the surface and then use the Tools - Make Image command.

We may also convert an image to a surface by copying the image and pasting it as a surface. The Paste as Surface dialog will appear when pasting that provides options for how the image data should be used to create a surface. Images that represent raster data such as elevations, temperatures and other factors are often very interesting to visualize as surfaces.

Linked Images from Image Servers

Linked Images from Manifold Image Servers

A recent development on the Internet is the emergence of a new class of image servers, which are web sites or web services that provide geographic images to users or programs seeking an image displaying a particular geographic region on Earth. Sites like TerraServer, Google Maps, Google Earth, Microsoft Virtual Earth, Yahoo! Maps and many more provide on-demand viewing of a variety of geographic images. Manifold can work with images from such servers if drivers are written that conform to a set of commonly-occurring interface standards expressed within the Manifold Image Server Interface API.

Depending upon the content provided by the server, such images can be:

- **Satellite or Aerial Imagery** - Photographic images that show a particular region as seen from above. Examples include TerraServer and Microsoft Virtual Earth.
- **Street Map Imagery** - Images that show a raster rendering of what would otherwise be presented as a vector drawing, as if someone made a screenshot of a drawing or vector map in a GIS system. The Microsoft Virtual Earth web service provides such images. Some image servers can present street map imagery with the "white space" made transparent. These displays are convenient to use as layers above satellite imagery layers so that street annotations can be seen over the satellite photos.
- **Other Images** - Images that show shaded-relief surfaces, renderings of CAD drawings (like a screen shot of a CAD drawing) and other content. Examples include images served by Manifold System web services.
Although the interfaces to such image servers vary in the details, in general they tend to have such similar capabilities that it is possible to use a more or less standardized dialog to link images from such servers. Manifold provides a standard Image Server Interface (ISI) that can be used to write an ISI driver for an image server so that it may be used from the standard Manifold image server dialog.

Image servers supported by an ISI driver are called Manifold Image Servers. ISI drivers can be either built into Manifold System or can be loaded as external add-ins. In either case, if an ISI driver exists within Manifold then access to the supported image server will appear to be a built-in Manifold capability whether or not the ISI driver was installed as an add-on or was built into Manifold itself.

By default, there are two Manifold Image Servers supported with ISI drivers:

- TerraServer - An ISI driver built into Manifold allows linking images from a variety of TerraServer themes.
- Manifold IMS Web Sites - A built-in ISI driver allows linking images from an IMS web site created using Manifold System if that web site was created with the IMS option to create an image server page. Note that using Manifold IMS to serve images in this way is a slightly different technology than the default IMS server and is in addition to Manifold's ability to function as an Internet Map Server or as an OGC WMS server.

In addition to the above two built-in ISI drivers, there are many other drivers created by third parties and distributed as open source projects, for example:

- Google Maps - An open-source ISI driver that allows using Google Maps as a Manifold Image Server to provide images of street maps. Must be installed.
- Google Earth - An open-source ISI driver that allows using Google Earth as a Manifold Image Server to provide satellite images and overhead aerial photographic images. Must be installed.
- Virtual Earth Street Map Image - An open-source ISI driver that may be downloaded from numerous sources on the web allows using Microsoft Virtual Earth as a Manifold Image Server to provide images of street maps. Must be installed.
- Virtual Earth Satellite Image - An open-source ISI driver that may be downloaded from numerous sources on the web allows using Microsoft Virtual Earth as a Manifold Image Server to provide satellite images and overhead aerial photographic images. Must be installed.
- Yahoo! Maps Street Map Image - An open-source ISI driver that may be downloaded from numerous sources on the web allows using Yahoo! Maps as a Manifold Image Server to provide images of street maps. Must be installed.
- NearMap - A super high quality image server providing satellite and street views, primarily in Australia.
- OpenStreetMaps image servers - Open source, open content street data presented by renderers such as CloudMade and others.

All of the above image servers must be installed. They are easy to download from the manifold.net image servers page (find it by drilling down through the Product Downloads page), except for the Google servers which are not distributed by manifold.net but which must be downloaded from other web sites.

Many image server providers will provide both satellite views and street map views. For example, the Microsoft Virtual Earth open source project provides access to both images of street maps and also of satellite imagery. manifold.net encourages all Manifold users to employ Microsoft Virtual Earth in preference to Google image servers. See the Linked Images from Google Servers topic for reasons why.

ISI drivers created by third parties are not supported by Manifold tech support. Such modules are often open source projects with full source code available so that communities of users can support each other and make whatever modifications are desired. Users often discuss such open source projects in the usual open source web sites or on Manifold user community forums.

Developers wishing to create ISI drivers for other image servers can study the open source code for existing image server modules (downloadable from the manifold.net web page) to learn how to create such drivers. Source code for such open source projects is available in the usual web sites serving the open source community. An ISI driver is, in effect an alternative browser that can connect to image server sites and provide a bridge for applications like Manifold to use the resulting imagery.
All Manifold Image Servers supported by ISI drivers will automatically fetch images to tile a particular region of interest as indicated either by the context of whatever component had the focus when the File - Link - Image dialog was launched or by manual specification of latitude and longitude extents, which can be a global extent if desired. Manifold Image Servers also normally allow a choice of resolutions and image levels to be downloaded if a linked image is to be converted into a static image within the project. The image linking dialog for all image servers using ISI drivers is virtually the same.

Very important: Before using an image server you must install the image server module for the image server you wish to use in your Manifold System installation. See the instructions below.

To create a linked image from a Manifold Image Server:

1. Image server modules will normally link worldwide coverage by default. If you want the image server to show only the region covered by a specific drawing or other component, open a drawing or other component showing the region to be covered by the image fetched from the server. Although we can specify the whole world or manually specify the latitude and longitude extents to be covered by the image, it is sometimes desired to let Manifold automatically restrict the view fetched to the extents of a given component. If you want to do that, open the component in a window and pan and zoom it to the region you want linked from the image server.
2. Choose File - Link - Image
3. In the Link dialog’s Files of type box choose Manifold Image Servers ()
4. In the resulting dialog, the Server box lists all Manifold Image Servers for which ISI drivers have been added to the system, plus the built-in setting for Manifold IMS Web Site. The built-in ISI-supported option of TerraServer will not appear in this dialog as it is accessed through its own Files of type choice. If you have installed any additional ISI drivers, such as those for Microsoft Virtual Earth, these will appear in the dialog. Choose the type of image server desired, for example, Virtual Earth Street Map Image.
5. Enter a URL to use for the connection (required for some servers, such as Manifold IMS web sites) or use the default URL provided (in the case of Virtual Earth or similar). To use a URL other than the default, click the Use custom server URL and enter the desired URL and then press the Refresh button. Some image servers, like those often provided for use with CloudMade allow you to change the formatting of the images based upon small changes in the URL, so you would click the Use custom server URL and enter the desired URL for those.
6. Specify the desired image Scale. Most people will use the default, which is usually the highest resolution available for that image server.
7. Choose the latitude and longitude extents to be covered by the image. To automatically get a worldwide image, press the Global button. If a component was open when File - Link - Image dialog was launched you can press the Window button to load the Longitude and Latitude range boxes with the correct latitude and longitude ranges to cover the opened component. If we would like to manually specify the latitude and longitude extents of the image we can do so.
8. Press the Refresh button next to the URL box. When the Refresh button is pressed, Manifold will establish a connection with the image server. This is normally quite fast, but may take tens of seconds or even longer. When the connection is established the OK button will be enabled. Press OK.
9. An image using a linked image icon will appear in the project. This image can now be opened and used like other images, except that images linked from image servers are read-only and cannot be re-projected.

When an image is linked from an image server, until the image is completely fetched, it will appear in an image window as a partially transparent gray rectangle covering the extents of the image. If the image is used as a layer in a map, it will appear in transparent color in those parts not yet fetched as tiles from the image server.

Images linked via the image server interface will use whatever coordinate system (projection) is returned by the image server module in use. If an invalid coordinate system is returned when linking an image Manifold will use the default coordinate system.

See the specific topics on liking images from the various ISI servers for details on using the linked image dialog with those servers.

Typical Link Image Server Dialog Controls

Manifold image servers will usually use a dialog with the following controls.
The server for the type of image desired. Different servers offer a variety of choices, such as different resolutions or color vs. grayscale images.

Use custom server URL
Specify the URL to use for the image server. If checked, enables specification of a custom URL in the Server box. This allows a change of URL if the image server moves from the default URL.

URL
URL to be used to connect to the image server. Enabled if the Use custom server URL box is checked.

Refresh - Establish connection to the image server.

Scale
Desired scale of the image. Some image servers provide a choice of different scales. Choosing 10m (ten meters) will result in a smaller image with fewer pixels than an image obtained by choosing 1m (one meter).

Longitude / Latitude
Longitude and Latitude extents of the image to be linked. Pre-loaded with values for worldwide coverage, like pressing the Global button by default.

Global
Press this button to automatically load the Longitude and Latitude boxes with values from minus to plus 180 degrees longitude and from minus to plus 85 degrees latitude. This provides worldwide coverage, with the most extreme Northern and Southern five degrees clipped to avoid problems with some third party image server modules which have problems very near the North or South pole.

Window
Press this button to automatically load the Longitude and Latitude boxes with values that cover the extents of whatever open component has the focus when the dialog is launched.

Cache data between sessions
Controls whether the cache folder used to cache image tiles retains its content between different sessions of Manifold or not. By default, the option is turned on. The cache folder is whatever location is set for Data Cache in the Tools - Options - File Locations dialog.

Proxy
Calls up the Tools - Options - Proxy server dialog that allows configuration of a proxy server connection, if a proxy server is used to connect to the Internet. Passwords supplied for proxy servers will be stored in encrypted form.

The Manifold Image Server Interface (ISI)

Manifold’s Image Server Interface (ISI) is an open standard published by manifold.net to make it easy for third party developers to create drivers that connect image servers to Manifold System. Image servers that can work with Manifold System using an ISI driver are called Manifold Image Servers and can be automatically accessed as if they were a built-in part of Manifold System using the File - Link - Image dialog.

See the Image Server Interface topic for a description of the interface. Programmers should also visit the manifold.net website for any supplemental documentation or example implementations, if any are made available in the future.

Third party developers creating image servers are strongly urged to craft an interface that, like those used by TerraServer, Microsoft and Yahoo!, is easy to support with an ISI driver. That will enable Manifold users to work with your image server from within Manifold System GIS projects. Using ISI provides maximum speed just as if your connection to the image server was built into Manifold’s own internal code.
To write a new ISI driver for your image server, see the open source projects for the ISI drivers for the existing image servers. These ISI drivers provide the open source community with a rich, practical example of how to write an ISI driver for a real-world image server. See the usual open source sites for access to the ISI drivers for Manifold image servers.

Creating ISI drivers for image services as an open source project also has the benefit that if these web sites change interfaces then the open source community can alter the ISI drivers without having to wait for some vendor, such as manifold.net, to issue a new release of software. This enables innovative companies like the image server providers to keep innovating while the open source community can continue working within open standards to support those innovations.

**Typical Image Server Driver Installation**

A typical image server driver installation package consists of a zip file that contains one or more .dll files. The file should be unzipped and the resultant .dll files should be copied into the Manifold System installation folder. When Manifold is launched it will detect the new .dll files and will add the supported to the image server dialog choices.

The default Manifold System installation folder is C:\Program Files\Manifold System. If you are operating a 64-bit Manifold installation in 64-bit Windows or a 32-bit Manifold installation in a 64-bit Windows system there will also be a second Manifold System installation folder at C:\Program Files (x86)\Manifold System. In that case, the .dll files for image server modules should be placed in both the C:\Program Files\Manifold System and also in the C:\Program Files (x86)\Manifold System installation folders.

After installation the .dll files will appear as seen in the above screenshot, showing how image server files for Google, Yahoo! and Microsoft Virtual Earth have been added to the default image server .dll files installed with Manifold.

**Projections and Linked Images from Image Servers**

Most image servers serve images using Mercator projection or a projection derived from Mercator. Images linked from image servers cannot be re-projected so if they appear as a layer in a map the map must use the same projection as the image. This is easy to guarantee by always creating a map component from the image linked from an image server.

Never create the map first and then drag and drop the image into the map. Instead, always right click on the image server image in the Project pane and then choose Create - Map in the context menu to create the map from the image server image. That will guarantee that the map is created using the same projection as the image server image. You can then drag and drop other components, like drawings, as layers into the map.

Although most image servers use Mercator projection, not all do, or some use a projection incompatible with Mercator. Image servers which use different projections cannot be used together as layers in a map, because if the map is created using the projection of one of the image servers the other image server image cannot be shown in that map. Therefore, although images linked from most image servers can appear together as layers in a map some might not be useable in that way.

Smaller images (limited by the Window button to smaller regions and thus smaller sizes) can be unlinked and then used as ordinary, local, unlinked images. Those may be re-projected into any projection desired. Very patient users can use this process to create remarkably large images if they have the free memory and disk space on their computers. For example, early research for the next generation of Manifold products used images 20 GB in size that were created in Release 8.00 by unlinking images originally linked from image servers.

Manifold provides a special datum for use by ISI driver writers, the **WGS84 Auto** datum. This is equivalent to the **WGS 84** datum used by Google and also used by Virtual Earth and Yahoo! Maps.

**Exporting Image Server Images**
An image linked from an image server may be exported to a file using the **File - Export - Image** command. The image linked from Virtual Earth will be exported as a single image using the specified image format.

If you have Enterprise Edition installed, an image linked from an image server can be shared to an Enterprise Server.

**Unlinking Image Server Images**

An image linked from an image server may be transformed into an ordinary image by right clicking the image linked from the image server in the project pane and choosing **Unlink**.

**Caution:** Images linked from an image server can be enormous and be far larger than can fit into either free disk available or memory on your computer. Think carefully about the size of image being unlinked, as it is very easy to carelessly ask for an image that is 0.3 meter in scale that covers such a large area that the resultant image requires terabytes of memory. Don't try this with a worldwide image linked from an image server. Instead, link an image using the **Window** button to constrain the image to a small, manageable extent and then convert, unlink or download it.

**Converting Image Server Images**

Another way to convert an image linked from an image server into an ordinary image is to open the image linked from the image server in a window and then choosing **Image - Convert**. Converting an image linked from an image server creates a local image in the project of the desired type, being equivalent to first unlinking it and then converting the new local image into an image of the desired type.

**Image Download**

The **Download** command appears when the focus is on a window for an image linked from an image server. The command allows automatic download of all tiles available for the image at any level of resolution available. Tiles will be assembled into a single image at each resolution level selected.

See the **Image - Download** topic for details.

**Notes**

Most users of image servers will work with global settings. Hard disk capacity is inexpensive so using the cache option (on by default) will build up a collection of local tiles for that image server that will be recycled by any project that uses those tiles. That leads to faster performance since tiles already fetched don't need to be re-fetched through Internet. Even if you have a fast Internet connection the image server you may be using is probably not as fast as access to your local hard disk.

Limiting extents using the **Window** button to only the extent of some previously opened component can be useful but it can also be frustrating if you want to pan the image server to a slightly larger view that wasn't part of the originally defined extent. No massive amount of detailed tiles will get fetched if you don't actually zoom in to a particular region, so there's no harm in working with **Global** extents. Because image servers use pyramided imagery only a very small amount of image pixels are involved in getting worldwide views when zoomed out.

Usually the main reason someone wants to work with a **Window** limitation is to have a view that's already zoomed in to a particular image server where navigation down to the desired small region from a global view would be tiresome. For example, suppose we have a drawing that shows a development of twenty houses in a small town and that is all we are interested in. We would normally zoom out the drawing a bit to provide some extra room on the margins and then link an image server using the **Window** button to restrict the image server linked image to only the view shown in the drawing. If we pop open the image server linked image right away we see that it is zoomed into the small region of the housing development. No need to zoom in from a global view.

We can then create a **map** component from the image server linked image and drag and drop the drawing as a layer into the map and see the two components together as layers in the map. Always create a map from the image server linked image to be sure that the projection of the map allows showing the image.

A project that uses an image linked from an image server may be saved within **.map** project file just like any other project. However, keep in mind if sending that **.map** file to someone else they also will need to have installed on
their machine the required image server module (if it is not one of the built-in image servers) to view the linked image. Loading a .map file with an image linked from a Manifold image server that is not installed on the system will report the problem and the name of the missing image server in the History pane.

**Cautions**

It is easy to think that we can avoid the effort of creating cool maps for all purposes by simply downloading them using an image server choice in the Server box. Although the use of image servers is indisputably convenient, the use of large raster images downloaded from the web is far from being a universal replacement for informed use of vector drawings.

Linking an image from an image server **does not** create a vector drawing, but instead links an image that shows a rasterized view of a nicely-formatted vector map. It can be a very bad idea to attempt to replace a vector drawing with an image of a vector drawing, as the image, if unlinked, may require hundreds of times as much storage space (or if not unlinked, bandwidth) as the equivalent vector map. Image servers may have licensing restrictions, and they may not be available when you want them.

For these reasons, no matter how convenient we find image servers for some purposes we should remember that image servers are not usually an efficient way of getting information into local storage for a GIS project. A better alternative can be to find a drawing that shows the same information (usually easy in the US) and then format it in a pretty way. The result will be a far smaller, more accurate data set that loads and displays much faster and which we entirely control.

On the other hand, using image servers to acquire satellite imagery and street map images for a particular region is often a very good idea, especially outside of the US where vector drawings may be extremely difficult to obtain. In many parts of the world the only way to get a good vector map is to trace it yourself from an image server image in Manifold.

**See Also**

- Image Server Interface
- Intermediate Levels and Pyramids
- Linked Images
- Linked Images from Manifold IMS Web Sites
- Linked Images from OGC WMS Servers
- Linked Images from TerraServer
- Working with Large Images or Surfaces
- Exporting KML to Google Earth
- A Flashy Demo - Web Queries and KML
- Fun with Google Earth
Linked Images from Manifold IMS Web Sites

Manifold can link images into a project using imagery streamed from a Manifold Internet Map Server (IMS) web site operated as an image server. The images so obtained will be the same as would be displayed for the requested location and zoom in an IMS web site.

This topic assumes the reader is familiar with creating Manifold IMS web sites.

To create a Manifold IMS web site for use with image server clients:

1. Create a Manifold IMS page in the usual way using the File - Export - Web Page dialog. In that dialog, make sure to check the Include Manifold Image Server interface page box.
2. This will create an imageserver.asp page within the web site.

To create a linked image from a Manifold IMS web site:

1. Open a drawing or other component showing the region to be covered by the image fetched from the server. Although we can specify manually the latitude and longitude extents to be covered by the image, it is most convenient to let Manifold do so automatically based upon the extents of the opened component.
2. Choose File - Link - Image.
3. In the Link dialog's Files of type box choose Manifold Image Servers.
4. In the resulting dialog, the Server box lists all Manifold Image Servers for which ISI drivers have been added to the system, plus the built-in setting for Manifold IMS Web Site. The built-in ISI-supported option of TerraServer will not appear in this dialog as it is accessed through its own Files of type choice. If you have installed any additional ISI drivers, such as those for Google Maps or Google Earth, these will appear in the dialog. Choose Manifold IMS Web Site as the type of image server desired.
5. Enter the URL for the Manifold IMS web site's image server .asp page, for example, something such as http://www.mysite.com/imageServer.asp and press the Refresh button to connect to that URL. [The example URL given in this step is just a made-up example: use whatever is the actual URL for the site you wish to access.]
6. When the Refresh button is pressed, Manifold will establish a connection with the Manifold server. This is normally quite fast, but may take tens of seconds or even longer. When the connection is established the OK button will be enabled and the size of the image at the requested Scale and Longitude and Latitude range will be reported in the number of pixels. To get a larger or smaller image, change the Scale value and press the Refresh button again.
7. If desired, choose the latitude and longitude extents to be covered by the image. If a component was open when File - Link - Image dialog was launched the Longitude and Latitude range boxes will be pre-loaded with the correct latitude and longitude ranges to cover the opened component. If no component was open or if we would like to manually specify the latitude and longitude extents of the image we can do so. Press OK.
8. An image using a linked image icon will appear in the project. Highlighting the image in the project pane will report it as an Manifold image in the status bar. This image can now be opened and used like other images. Manifold IMS web site image server images are read-only and cannot be re-projected. Until the image is completely fetched, it will appear in an image window as a partially transparent gray rectangle covering the extents of the image. If the image is used as a layer in a map, it will appear in transparent color in those parts not yet fetched as tiles from the Manifold server.

Logotype

If the "Powered by manifold.net" logo is turned on in the IMS web site, then each image will be stamped with that logo. Enterprise Edition or higher Manifold edition must be used to turn off the logo.

See Also

Linked Images
Linked images from Manifold Image Servers
Linked Images from Google Servers
Linked Images from OGC WMS Servers
Linked Images from TerraServer
**Linked Images from Google Servers**

Google provides web services that allow download of images showing maps like those in Google Maps and satellite images like those seen in Google Earth. Both images showing maps and images showing satellite images are often referred to as Google Maps data even though the satellite images are normally viewed within the Google Earth browser.

Manifold can link images from Google image servers if we have an Image Server Interface (ISI) driver to use the Google servers as Manifold Image Servers installed on our machine. See the Linked Images from Manifold Image Servers topic for information on the image server interface.

**Manifold.net** does not distribute image server modules for Google; however, many third parties do. A quick web crawl is usually all it takes to locate a Manifold image server module that conforms to the Manifold ISI standard and works with Google servers. For maximum irony, use Google to search the web to find image server modules for Google. If you want to be mean to Google, use Bing.

**Note**

**manifold.net** encourages all Manifold users to employ Yahoo!, Microsoft Virtual Earth, CloudMade, Yandex or NearMap in preference to Google image servers.

There are many important reasons why using Microsoft Virtual Earth ISI modules is a better idea than using Google Earth ISI modules:

- Manifold users who have experience with Google ISI connections have complained that Google has an undocumented "throttle" of some sorts that limits access to Google servers if people actively use the connection. Microsoft Virtual Earth has no such throttle and encourages use.
- Users with experience of Google ISI drivers complain that Google-sourced imagery sometimes is stamped with annoying "watermarks" that say "Google" over the image. Microsoft Virtual Earth does not disfigure the imagery it provides.
- Microsoft Virtual Earth imagery appears much smoother than Google imagery at large scales, without the annoying "mosaic" tile effects often seen in Google imagery at large scales.
- In some cases, Microsoft Virtual Earth imagery is more detailed at local scales than Google imagery.
- Microsoft provides an extraordinary range of developer-friendly programs to assist the use of Virtual Earth, either using Microsoft APIs or using open source connections unregulated by Microsoft.
- Google has attempted to impede developers providing open source connections to public Google web services. Microsoft has supported and encouraged such developers. Understanding and applying Google's restrictive legal positions for its API requires costly legal investigation, while Microsoft makes it very easy from a legal and business perspective to utilize Microsoft Virtual Earth.
- Numerous third party geographic browsers, like NASA World Wind, have connections to Virtual Earth while Google has tried to prevent competition from such browsers. Virtual Earth is therefore becoming the standard for geographic browsers.
- All Manifold development is aimed at Virtual Earth and Virtual Earth will be supported in future editions of Manifold directly by manifold.net. Google Earth is supported only by third party developers via open source projects that may or may not continue to be available with no development or technical support for such projects or modules by manifold.net. By georegistering your work to Virtual Earth you increase the likelihood that no adjustments will be required (in georegistration or otherwise) or that fewer adjustments will be required with future products from manifold.net.

For the above reasons, most experienced Manifold users working with ISI drivers have already switched to Microsoft Virtual Earth.

Other image servers popular with Manifold users include Yahoo! and newer entrants like CloudMade, which provide far greater control over display than Google. Yahoo! in particular is to be commended and supported for their very generous encouragement of the OpenStreetMaps project.

**See Also**

Linked Images
Linked images from Manifold Image Servers
Linked Images from Manifold IMS Web Sites
Linked Images from OGC WMS Servers
Linked Images from TerraServer

A Flashy Demo - Web Queries and KML
Exporting KML to Google Earth
Fun with Google Earth
Export Drawing - KML, KMZ
Export Image - KML, KMZ
Import Drawing - KML, KMZ
Linked Images from OGC WMS Servers

An OGC WMS server is one that can communicate using "Open" GIS Consortium Web Mapping Service (WMS) or similar communications specification. A WMS server produces an image of GIS data and streams the image to requesters. If the original data is vector data such as a drawing, the WMS server does not provide the drawing data, it only provides an image view of the data in some image form like a GIF or JPG.

Manifold is capable of linking images from WMS servers. Manifold will automatically negotiate with the WMS server to achieve a mutually understood level of the various WMS specifications that have been published. When operating as a WMS client Manifold will automatically correct several types of malformed URLs that may be returned by defective servers.

Manifold can also function as a WMS server as part of Manifold's Internet Map Server capabilities. See the Map Server Overview topic for additional information on running Manifold as a WMS Server.

WMS servers create an image at the desired size and then stream views of that image in tiles to the client depending upon the view position and the zoom desired. As we zoom farther into the image more tiles of greater detail are sent up to the limit of resolution implied by the specified size of the image. Note that the original choice of image size implies a limit of resolution per pixel implied by the extent of the original data divided by the number of pixels.

Manifold will cache image tiles streamed from an OGC WMS server so that panning and zooming in the image can be as fast as possible. For example, when zooming into an OGC WMS image Manifold will make use of less detailed image tiles that have already been fetched to quickly render the desired view as best as possible and then progressively update the display as more detailed tiles stream in.

Images linked from WMS servers are read-only images and cannot be re-projected. They can only participate in maps that use the same projection as the image, although they may be used in maps that have the same projection (coordinate system) except for the datum used. At any time, an image linked from a WMS server may be unlinked to create a local image that may be re-projected and otherwise altered as desired.

An image linked from an image server may be converted to a local image within the project either by using the Image - Unlink command or the Image - Download command. The Unlink command converts the existing linked image to a local image at whatever resolution is currently in use. The Download command may be used to automatically fetch all image tiles at any available resolution to create a local image at that resolution.

Images linked from a WMS server may also be exported to image files. In that case, the exported image will be the same as that which would be obtained by unlinking the WMS image.

Note that exporting or unlinking an image linked from a WMS server will only use the data already downloaded from the server and will not attempt to download any more data. It will also only use the data for the most detailed image level. It is usually a good idea to use Image - Download to check how much data has already been downloaded for the most detailed image level prior to exporting or unlinking the image.

If the connection is severed between a linked image in a project and the image server (such as, for example, if the project is moved to a different machine and a local URL must be changed to a global URL) the link to the image server may be re-established using the Image - Relink command.

Another way to create a local image from a linked image is to open the linked image and then to use the keyboard F6 command or the Tools - Make Image command. The Make Image dialog allows us to create a local image in several different ways from the linked image without unlinking the linked image. For example, we can use this command to create an image of the displayed window or of the entire linked image at a desired number of pixels or scale.

To create a linked image from an OGC WMS Server:

1. Choose File - Link - Image
2. In the Link dialog's Files of type box choose OGC WMS Servers
3. In the Link OGC WMS Data dialog enter a connection string for the Server and press the Refresh button in the dialog's toolbar.
4. When the Refresh button is pressed, Manifold will communicate with the WMS server to find out what layers are available. This can take a very long time, minutes at a time, in the case of slow WMS servers. When the catalog of layers (the "Capabilities" in WMS server jargon) has been fetched, it will be displayed in the Layers pane.
5. Specify the desired image **Size**, select the desired **Layers** and the desired **Style** for each layer and then press **OK**. Choosing more than one layer tells the WMS server to create an image that combines all selected layers into one image.

6. An image using a linked image icon will appear in the project pane using a name based on the name of the WMS server. Highlighting the image in the project pane will report it as an **OGC WMS** image in the status bar. This image can now be opened and used like other images. **OGC WMS** images are read-only. Until the image is completely fetched, it will appear in an image window as a partially transparent gray rectangle covering the extents of the image. If the image is used as a layer in a map, it will appear in transparent color in those parts not yet fetched as tiles from the WMS server.

**Link OGC WMS Data Dialog Controls**

- **Server**: URL to be used to connect to the WMS server. The **Server** combo remembers recently used WMS servers, including those used in earlier sessions of Manifold.

- **Size**: Desired size of the image. For maximum efficiency, use image sizes that are evenly dividable by the **Tile Size**.

- **Tile Size**: Size of tiles to use when fetching the image, ranging from **64 x 64** pixels to **1024 x 1024** pixels. Larger sizes take longer to download but use storage space more efficiently.

- **Refresh**: Update the **Layers** pane with layers available from the WMS server.

- **Select All**: Check all layers for retrieval.

- **Select None**: Uncheck all layers.

- **Select Inverse**: Uncheck all checked layers and check all unchecked layers. A fast way to retrieve all but one layer: click **Select None**, check the one layer not desired and then click **Select Inverse**.

- **Layers**: After the **Refresh** button is pressed, the **Layers** pane shows images available from the WMS server. The **Style** option chooses which optional style will be used.

- **Background color**: Controls the background color of the image. If the server can serve images in PNG, GIF or other format that retains invisible pixels, the option is turned off (but can be turned on). If the server cannot serve images in a format that retains invisible pixels, the option is turned on and cannot be turned off.

- **Cache data between sessions**: Controls whether the cache folder used to cache image tiles retains its content between different sessions of Manifold or not. By default, the option is turned on.

- **Link each layer as a separate image**: If unchecked (the default) all layers will be composed into a single image. If checked, all layers will be linked as separate images in the Manifold project. If the **Link OGC WMS Data** dialog is used to relink an existing image, the option is disabled (and turned off).

- **Proxy**: Calls up the Tools - Options - Proxy server dialog that allows configuration of a proxy server connection, if a proxy server is used to connect to the Internet. Passwords supplied for proxy servers will be stored in encrypted form.

Most WMS servers cannot provide a default image size, so the **Link OGC WMS Data** dialog provides a **Size** control to specify the desired size of the image. The larger the dimensions, the greater the resolution per pixel and thus the deeper we will be able to zoom into the image. The default dimensions are **4096 x 4096**, which is a good...
Images

A tradeoff between deepness of zoom and the amount of data which must be fetched from the server to display the entire image at high zoom.

Images are served by WMS servers as tiles. Tiles may range in size from $64 \times 64$ pixels to $1024 \times 1024$ pixels. The default setting of $256 \times 256$ is a good tradeoff between efficiency and flexibility.

Because images are fetched as tiles, the default dimensions for overall image size are slightly uneven ($4096$ instead of, say, $4000$) because the image is fetched in tiles that are $256 \times 256$. Dimensions of $4000 \times 4000$ would work, but would not be as efficient because the overall size dimensions are not evenly divisible by the tile size.

Layers displayed in the Layers pane are arranged according to their hierarchy on the server. Layers that contain no data and are used to group other layers (virtual layers) use gray checkboxes. Layers that contain real data (data layers) use black checkboxes. The total number of checked data layers is shown in the status readout below the layer tree.

User Names and Passwords

When connecting to some image servers, the URL given must include a user name and a password in the URL string. Manifold supports a common way of doing so using URLs in the form of:

```
http://user:password@host
```

For example, if our user name was johnbgood and our password was $3bart8tt2$ and the hostname for the image server was imagesupermarket.com, we could use a URL connection string of:

```
http://johnbgood:3bart8tt2@imagesupermarket.com
```

Note that the above style is just one popular way webmasters arrange to accept user login names and passwords as part of an image server URL. There are various other methods in use. If the image server we work with uses the above method we are in luck.

Layer Styles

Some WMS servers may offer to display a given layer using one or more optional styles within the synthesized image. If so, the Style cell for that layer will be a combo that allows choice.
For example, in the above illustration the Crosshairs layer has a variety of styles advertised by the server. We can change the style from the default red wire to blue wire if we like.

Not all layers will have a Style option. What styles are available for any given layer will be up to the WMS server providing the data, not Manifold.

**Fetching Image Tiles from a WMS Server**

Consider the following scenario: We link an image from a WMS server and a new linked image icon appears in the project pane. We open the image in a window zoomed to fit. Manifold will paint the area occupied by the image in semi-transparent gray and will start retrieving the image tiles required for the current view from the WMS server. As tiles are retrieved, Manifold will update the display.

Because image tile retrieval occurs in a background thread, we can continue doing other work as the image assembles from tiles retrieved from the WMS server. If the WMS server is fast and our Internet connection to the WMS server is fast, the image will assemble much faster than if the WMS server is slow or if our Internet connection is slow.

Suppose we zoom in to some region within the image, which we can do before all tiles have been fetched. Manifold will immediately begin retrieving tiles for the new view, which take priority over tiles in the old view. When Manifold retrieves all tiles in the new view, it will go back and continue retrieving any tiles in the old view that were not fetched when we zoomed in. Manifold does this so that the old view is ready for use just in case we decide to return to it later, say, by using View - Go Back. If we close the project file or delete the image the system will stop retrieving tiles from the server.

Tiles are retrieved one at a time, but tiles for several images can be retrieved simultaneously and more than one window containing an OGC WMS image can be open. The active window has elevated priority when rendering OGC WMS images in more than one open window. If we display an OGC WMS image in two windows (for example, in two maps), Manifold will first try to retrieve tiles for the active window, that is, the window that has the focus, and then for the window in the background.

Note that some WMS servers are run as advertising functions for the company providing the data shown in the images. WMS servers are often configured to mark each tile they send with a copyright text, a logo or some other
mark. If this is the case the mark will be repeated on each tile. Do not blame Manifold for this behavior as it is the WMS server that is disfiguring the tiles.

**Cache Files**

Manifold caches tile files retrieved from the WMS server so that panning or zooming to previously-viewed scenes does not require a re-fetch of previously retrieved tiles.

The **Cache data between sessions** option controls whether or not cached tile files will be retained in the cache folder between different sessions of Manifold. By default, the option is turned on.

The location of the cache folder is specified by the **Data Cache** item in **Tools - Options - File Locations**. The default value for the location is `%MyDocuments%`, so by default each user's files will be stored in the user's own cache folder. If we actually use the **My Documents** folder for other documents, we might not want to have many cache files appear in the same folder as various other working documents we use.

In that case, it is a good idea to create a folder within the **My Documents** folder, perhaps called **Cache** and then we can set the location for the data cache to `%MyDocuments%\Cache`. This will keep all of our WMS cached files in a separate folder so that we will not have to wade through long lists of cached files when working with other documents.

When multiple users are working it might make sense to modify the default **Data Cache** setting so the cache folder is on a shared resource available from all machines on the local network. For example, if there is a machine called **Storage** on our local network with a **Cache** folder on its **C**: drive, we might use a **Data Cache** setting of `\Storage\C\Cache` for all users. In that case, any tile files that are retrieved by one user will be available to all users so that tile files for a given view will never have to be fetched twice over the network.

Cache folders are organized as is as follows:

- Each combination of layers, styles and background color for a given URL is given a separate subfolder used to store PNG, GIF or other files containing individual image tiles.
- Names of tiles include the level and the XY coordinates of a tile.
- The overall dimensions of the image are in level 0. The dimensions in level 1 are half those of level 0, and so on.
- The subfolder containing the tiles also contains an XML file (currently named **RESOURCE.XML**) with the URL, layers, styles and background color used for the image.

**Caution:** Cache files persist forever and, in extreme cases, use up your free disk space unless they are manually deleted. See the recommended actions in the Managing Cache Files topic.

**WMS Version Negotiation**

Clicking **Refresh** the **Link OpenGIS WMS Data** dialog begins a negotiation with the WMS Server of the WMS protocol version to be used. Regrettably, the spastic approach used by OGC to design WMS has resulted in a plethora of irritatingly dissimilar WMS protocols that do pretty much the same thing. Manifold must negotiate with the WMS server to learn what protocol it uses. Manifold first tries to use the protocol version suggested by the WMS server. If the server refuses to suggest a WMS version or the version it suggests is not supported by Manifold, Manifold tries to use versions 1.3.0, 1.1.2, 1.1.1, 1.1.0 or 1.0.0. If the WMS server supports none of these versions, the connection fails. As the OGC bureaucracy dreams up yet more WMS variations, future versions and service packs of Manifold will add them to the roster of protocols to attempt.

The negotiation process initiated by clicking **Refresh** is done in a background thread to avoid locking the user interface.

Images linked from a WMS server will have their coordinate systems automatically adjusted by Manifold so the image has the same aspect in X and Y dimensions. This corrects images fetched from servers that do not conform to the WMS specification and thus distort the images they return (a surprisingly frequent problem).

**Projections and OGC WMS**
There is no standard way for an OGC WMS server to describe the coordinate systems (projections) it supports that covers all coordinate systems supported by Manifold. There are several semi-standard ways using schemas to describe the coordinate system used by a WMS server that cover various subsets of coordinate systems.

When used as an OGC WMS client, Manifold supports all schemas for describing coordinate systems that have been uncovered by manifold.net, no matter how many or how few OGC WMS servers that currently exist actually use them. Given the tendency of OGC to come up with new "standards" that are never used to any significant degree, it is always possible yet another schema may be introduced in the future that is not recognized by Manifold as an OGC WMS client; however, we feel confident that the current edition of Manifold when used as an OGC WMS client covers all such schemas either proposed or in actual use.

In contrast, when used as an OGC WMS server, Manifold only supports EPSG codes. EPSG is arguably both the richest and the most widely used schema within the OGC community. Manifold running as a WMS server only supports EPSG mostly to avoid confusing various non-Manifold WMS clients, some of which, unlike Manifold, cannot auto-adapt to a variety of coordinate system encoding schemas.

As rich as it is, the EPSG schema only supports a fraction of the coordinate systems that can be used in Manifold. If the coordinate system used by the component served by Manifold can not be expressed using the EPSG schema, that is, if it does not have an equivalent EPSG code, then the Manifold OGC WMS server reports the coordinate system as latitude / longitude and will re-project the data for that component on the fly into Latitude / Longitude. For large components that will have a terrible impact on performance.

Obviously, it is essential that any webmaster setting up a Manifold application as an OGC WMS server should check the projections used by the components being served and should make sure that they are coordinate systems that can be expressed using the EPSG schema.

The current list of coordinate systems which have EPSG codes may be obtained free of charge from the following link, clicking the link to the epsg-v69.zip or subsequently published file.:

http://www.epsg.org/CurrentDB.html

If all we are doing is using Manifold as an OGC WMS client we don't need to worry about the above because any schema used by the OGC WMS server we access as an image server will be recognized by Manifold.

**Most WMS Servers are Absurdly Slow**

In a perfect world, Internet connections to WMS servers will be very fast and images obtained from such servers will pop open with the click of a mouse. It may be possible to achieve such performance if the WMS server is an in-house machine serving private clients over a purely local network or over a private, high speed Intranet.

In the real world, WMS servers are often too slow to use for most practical purposes. There are two reasons for such incredible slowness: First, WMS servers tend to be very slow in their response to requests because, of course, like any popular site they are overloaded and the bureaucracies that run most of them tend to skimp on resources for public users. Second, it is almost a rule of nature that the Internet connection between our computer and the distant WMS server will be excruciatingly slow. Therefore, if we plan on using WMS servers we should get used to seeing semi-transparent gray images that get slowly assembled, tile by tile, into a final image.

Even if we have a fast link to Internet, such as a 6000/608 DSL connection, we must keep in mind that the rest of Internet and, in particular, the link to the WMS server can be very, very slow. With some slow servers it can take five minutes to fetch layers available once the Refresh button is pressed and then another five or ten minutes for an image to be populated. It only takes a few experiences with WMS servers to be very grateful that Manifold can download WMS image tiles in background while we continue on with other work.

Once the image has been fetched and browsed to whatever extents and zooms are needed then it will be saved in Manifold's cache. From then on, response will be fast as tiles are fetched from the local cache on our machine and not through an (incredibly slow) Internet connection.

**Notes**

A WMS server does not serve the original GIS data: it serves a snapshot of that data in the form of an image. To understand the difference, consider a map within a Manifold project that consists of a dozen layers, where each layer might be a drawing, surface or other component. At any time we can open the map and use the Tools - Make Image command to create an image from that map.
We can then open that image. However, the image we made from the map is not the map nor does it contain the vector data from that map. It's more like a screenshot. The image created and served by a WMS server is the same sort of image. It's just a "screenshot" at the resolution specified.

If the objective is to grab data that was originally in raster form, the use of WMS servers can be a very good idea and very useful. If the objective is to make authentic use of real data, then, unfortunately it is not such a good idea as raster images are very poor, at best partial replacements for bona fide vector data.

When used to serve images of vector data, the WMS server does not provide the original data to users in a form that is genuinely useful to GIS users. Instead, only a raster image representation of the original, true vector data is provided. Instead of getting access to the original data so the user can create his or her own maps, edit them, analyze them as they see fit and publish them as they would like, the user only gains access to that image view that the publisher deems appropriate to present.

When used in this way, WMS servers are therefore becoming popular with bureaucracies that would like to give the appearance of public access to data without actually providing any public access to the real data. This is a good example of how so-called "Open" GIS specifications are used in real life to assure "closed" systems and to enforce centralized control.

It is, of course, perfectly legitimate for private data owners not to want to publish the underlying data that is used to create WMS server images. We simply want to emphasize that publishing a WMS image is not the same as providing the underlying data, a distinction that many government WMS server sites fail to make.

There are many ways of using WMS servers. For example, Manifold itself can function as a WMS server. In theory, WMS servers can be a great way of creating background layers that can be varied, or to serve images or images of surfaces in a variety of applications. In the case of vector data, there's nothing wrong with using a WMS server to provide a handy source of images of that data (for people who do not have GIS capability). If desired, access to the underlying data can be assured by providing some means of easy download of the original vector data, such as an FTP or other link for downloads.

**OGC WMS Servers and ISI Drivers**

While most connections to image servers within Manifold (connections to TerraServer, Virtual Earth Maps, Virtual Earth Earth and Manifold Image Servers) are accomplished using Manifold Image Server Interface (ISI) drivers, the connection to OGC WMS servers does not use ISI drivers. OGC WMS servers are a standard unto their own and are sufficiently unlike other image servers (such as TerraServer or the Virtual Earth servers) that they cannot use ISI. Therefore, the internal Manifold connection to OGC WMS servers is specifically hard-wired for those servers.

**About ISI**

Manifold's Image Server Interface (ISI) is an open standard published by manifold.net to make it easy for third party developers to create drivers that connect image servers to Manifold System. Image servers that can work with Manifold System using an ISI driver are called Manifold Image Servers and can be automatically accessed as if they were a built-in part of Manifold System using the File - Link - Image dialog. See the Linked Images from Manifold Image Servers topic.

**See Also**

Linked Images
Linked images from Manifold Image Servers
Linked Images from Manifold IMS Web Sites
Linked Images from TerraServer
Image - Download
Image - Relink / Unlink
Images can be Inefficient
Tools - Make Image
Public Access to Public Data
GIS and Networking
Map Server Overview
Linked Images from Oracle Servers

Manifold System Enterprise Edition and higher editions can link images into a project that are stored within Oracle Spatial databases or within standard Oracle databases using the GeoRaster facility. If you do not have at least Enterprise Edition installed, you will not be able to use the features described in this topic.

Oracle data sources using Oracle Spatial or that include GeoRaster capability (available in some other Oracle DBMS products) can store images as well as drawings. Manifold can import images or link images from an Oracle data source into the Manifold project. Manifold can import a variety of image types from Oracle data sources, including indexed (palette) images and images stored with channel interleaving.

Images are imported or linked from Oracle using the Database Console dialog.

To link an image from an Oracle server using the Database Console:

1. Launch the Database Console using Tools - Database Console.
2. Connect to the Oracle data source desired.
3. Browse the data source to locate the image desired.
4. Right click on the image and choose Link to link it or Import to import it.

The image will appear in the project pane as a linked image.

Oracle Data Source Dialog Controls

Server URL to be used to connect to the Oracle server.
User name User name, if required to login to the server.
Password Password, if required to login to the server.
Test Press to test the connection to the server.

Oracle Images and Projections

Oracle images are delivered by the Oracle server in a specified projection, which Manifold will maintain. As a practical matter they cannot be reprojected. To use such images in a map, create the map based upon the Oracle image so the map uses the same projection as the Oracle image. Alternatively, Unlink the image so it becomes a local image stored within the project. We can then do with it whatever we please.

Exporting Oracle Images

An image linked from Oracle may be exported to a file using the File - Export - Image command. The image linked from Oracle will be exported as a single image using the specified image format.

An image linked from Oracle can be shared to an Enterprise Server.

Unlinking Oracle Images

An image linked from Oracle may be transformed into an ordinary image by right clicking the image linked from Oracle in the project pane and choosing Unlink.

Converting Oracle Images

Another way to convert an image linked from Oracle into an ordinary image is to open the image linked from Oracle in a window and then choosing Image - Convert. Converting an image linked from Oracle creates a local image in the project of the desired type, being equivalent to first unlinking it and then converting the new local image into an image of the desired type.
**Linked Images and Channels**

Images linked from native Oracle data sources allow custom selection of which Oracle channels should be used for R, G and B channels via the **View - Display Options** dialog.

![Display Options Dialog](image)

By default the image will use **Channel 3** for **Red** color, **Channel 2** for **Green** color and **Channel 1** for **Blue** color. If we check the **Custom channels** box we can change these settings to whatever we like. We might wish to change channel assignments either to create a "false color" image or to adjust channels to their natural correspondence if an image was stored to Oracle using unusual channel / color assignments.

**See Also**

- Linked Images
- Image - Download
- Image - Relink / Unlink
- Oracle Spatial Facilities
- Tools - Database Console
Linked Images from TerraServer

Manifold can automatically link images served by Microsoft's TerraServer image server. TerraServer is a free server operated by Microsoft Corporation that serves free aerial images and scanned USGS maps (DRGs) provided via a partnership with the U.S. Geologic Survey. At present, TerraServer provides images and raster maps for the United States, including Puerto Rico. See http://terraservice.net or http://www.terraserver-usa.com for details.

Important: The very cool Microsoft TerraServer should not be confused with the various non-Microsoft enterprises (such as terraserver.com) that have nothing to do Microsoft even though they have managed to obtain confusingly similar domain names.

TerraServer may be accessed either via an interactive web site visited by users, or as a web service to which applications like Manifold may connect to automatically fetch desired images.

- The Download and Mosaic TerraServer Images example topic shows how to work with images obtained using the interactive TerraServer website by hand and to manually assemble them into mosaics.
- This topic describes how to use the built-in Manifold capability of automatically downloading and combining images from TerraServer.

For most purposes, using the built-in Manifold ability to fetch and assemble TerraServer images is much more convenient than manually visiting the web site and then downloading and assembling images by hand.

Manifold can automatically access TerraServer to fetch desired images that cover a particular area of interest. When images are available for the area of interest, Manifold can automatically fetched the required TerraServer images and seamlessly mosaic them together into a single image.

If a component window is open and has the focus Manifold can automatically fetch and mosaic images from TerraServer to cover the region represented by that window (assuming, of course, the opened window is correctly georegistered and shows a region somewhere in the US). If a window is not open, we can still fetch images from TerraServer by specifying the desired latitude and longitude ranges of the region to be imaged.

To fetch an image from TerraServer, use the File - Link - Image command. In the Link Image dialog, choose TerraServer Server () in the Files of type box to launch the Link TerraServer Data dialog.

TerraServer images will be streamed from TerraServer as image "tiles" at the level of resolution required to construct the desired image. As we zoom into an image, Manifold will fetch more detailed tiles from TerraServer.

Manifold will cache image tiles streamed from an TerraServer server so that panning and zooming in the image can be as fast as possible. For example, when zooming into an TerraServer image Manifold will make use of less detailed image tiles that have already been fetched to quickly render the desired view as best as possible and then progressively update the display as more detailed tiles stream in.

Images linked from TerraServer are read-only images and cannot be re-projected. They can only participate in maps that use the same projection as the image, although they may be used in maps that have the same projection (coordinate system) except for the datum used. At any time, an image linked from TerraServer may be unlinked to create a local image that may be re-projected and otherwise altered as desired.

An image linked from an image server may be converted to a local image within the project either by using the Image - Unlink command or the Image - Download command. The Unlink command converts the existing linked image to a local image at whatever resolution is currently in use. The Download command may be used to automatically fetch all image files at any available resolution to create a local image at that resolution. Images linked from TerraServer may also be exported to image files. In that case, the exported image will be the same as that which would be obtained by unlinking the TerraServer image.

Note that exporting or unlinking an image linked from TerraServer will only use the data already downloaded from the server and will not attempt to download any more data. It will also only use the data for the most detailed image level. It is usually a good idea to use Image - Download to check how much data has already been downloaded for the most detailed image level prior to exporting or unlinking the image.

Another way to create a local image from a linked TerraServer image is to open the TerraServer image and then to use the keyboard F6 command or the Tools - Make Image command. The Make Image dialog allows us to create a local image in several different ways from the TerraServer image without unlinking the TerraServer
image. For example, we can use this command to create an image of the displayed window or of the entire linked TerraServer image at a desired number of pixels or scale.

If the connection is severed between a linked image in a project and the image server (such as, for example, if the project is moved to a different machine and a local URL must be changed to a global URL) the link to the image server may be re-established using the Image - Relink command.

To create a linked image from TerraServer:

1. Open a component window (such as a drawing or a map or other component) and pan it and zoom it so show the region you would like to have covered by TerraServer imagery.
2. Choose File - Link - Image
3. In the Link dialog’s Files of type box choose TerraServer ()
4. In the Link TerraServer Data dialog enter a URL to the TerraServer server in the Server box and press the Refresh button. The Server box will be pre-loaded with the URL of the default Microsoft TerraServer server, but it is possible that experts may wish to use different URLs.
5. If your connection to Internet goes through a proxy server, press the Proxy button and provide the required connection information for your proxy server.
6. Specify the maximum Scale of interest. Note that the Latitude and Longitude range boxes have been pre-loaded with latitudes and longitude values that precisely cover the extent of the view seen in whatever open component window has the focus. Press the Refresh button next to the Server box.
7. When the Refresh button is pressed, Manifold will communicate with the TerraServer server to find out what images are available for the region specified by the given latitudes and longitudes. This can take a very long time, possibly even a few minutes at a time, when TerraServer is busy. When the catalog of layers (the “Themes” in TerraServer jargon) has been fetched, all available images for the region of interest will be displayed as choices in the Themes box. For many parts of the US the choices are normally a DOQ, which is an overhead aerial image, or a DRG, which is an image created by scanning a paper USGS map. Some regions may have no images available. Choose a desired image in the Theme box.
8. Press OK. An image using a linked image icon will appear in the project pane using a name based on the theme, for example, “DOQ”. In a background process, Manifold will begin downloading the image into local cache so that it may be conveniently displayed.
9. Highlighting the image in the project pane will report it as a TerraServer image in the status bar together with the size of the image in pixels.

The new linked image can now be opened and used like other images except that until they are unlinked to create a local image, TerraServer images are read-only and cannot be re-projected. Until the image is completely fetched, when the image component is clicked open it will appear in an image window as a partially transparent gray rectangle covering the extents of the image. If the image is used as a layer in a map, it will appear in transparent color in those parts not yet fetched from TerraServer.

**Link TerraServer Data Dialog Controls**

- **Server** URL to be used to connect to the TerraServer server. The Server combo remembers recently used TerraServer servers, including those used in earlier sessions of Manifold.
  
  - **Refresh** - Update the Theme box with images available from the TerraServer server.

- **Theme** Images available for the specified region. Populated only after the Refresh button has been used to make contact with TerraServer.

- **Scale** Maximum resolution of the desired image expressed as the smallest discernable feature size. For example, 1 m images have pixels that are one meter in size.

- **Longitude** Longitudinal extent of the desired image. If a component window is open and has the focus these values will automatically be filled in to cover the view shown in the window.
Latitude

Latitudinal extent of the desired image. If a component window is open and has the focus these values will automatically be filled in to cover the view shown in the window.

Cache data between sessions

Controls whether the cache folder used to cache image tiles retains its content between different sessions of Manifold or not. By default, the option is turned on.

Proxy

Calls up the Tools - Options - Proxy server dialog that allows configuration of a proxy server connection, if a proxy server is used to connect to the Internet. Passwords supplied for proxy servers will be stored in encrypted form.

If no windows are open when the Link TerraServer Data dialog is opened, we will have to provide the latitude and longitude extents of the region for which we desire an image. If a window is open, then the latitude and longitude extents of that window will be automatically filled into the Latitude and Longitude boxes for the dialog. This is a great convenience but is subject to a few common-sense rules:

The open window must be a 2D graphical window, such as a drawing, theme, image, surface, labels component or map. The component must be correctly georegistered so that the window shows some real location on Earth. This is easy to verify: when moving the mouse cursor over the opened window the status bar will report the location of the mouse cursor. If the Tools - Options - Status Bar pane setting for Current location is selected and set to latitude / longitude the status bar will report the position of the mouse cursor in latitude and longitude coordinates.

If the latitude and longitude locations make sense (that is, if they are what we would expect them to be given the view shown in the window) then we know that the component in the window has, in fact, been correctly georegistered. If on the other hand the locations are obviously wrong (say, as if we are looking at an image showing an aerial photograph of Chicago and the locations are reported as some spot in the ocean off the coast of Africa) we know that the component is not correctly georegistered. This usually happens when a component from a non-geographically aware format has been incorrectly imported.

See the Projections and Imported Components topic for information about correctly importing components from formats that do not save georegistration information.

The Scale Parameter

The value of the scale parameter determines the size of pixels in the Manifold image that is created from TerraServer data served into the link. For example, a scale parameter of 1m will create an image where each pixel is one meter in size. Together with the bounding box of the image (that is the geographic extent of the image), the pixel size determines the dimensions of the image component in number of pixels vertically and horizontally. The dimensions and pixel size of the image are used when exporting the image and in other operations (such as Image - Download) done with the image.

The value of the scale parameter represents the maximum scale used by Manifold when downloading data from TerraServer. As we zoom in and out in the image, Manifold will switch display scales in order to download as little data as possible from TerraServer (fetching only enough data to fill the display window at the current zoom), except that it will never use a scale that is more detailed than the scale parameter specified when linking the image. Zooming in beyond that scale will result in pixels “expanding” into larger block shapes just as occurs when zooming far into an ordinary, local image.

Thus, the scale parameter allows us to limit the amount of data with which we work. This is an important consideration when working with TerraServer data, since requesting TerraServer coverage over a large geographic region can involve an immense amount of data at the highest resolution TerraServer can provide. By asking for a larger scale, we can still image a large geographic region at a convenient level of detail without waiting eons for the download of lots of fine detail that is not necessary.

Cache Data Option

The Cache data option is important because it keeps a copy of the linked image in local disk cache. If this option is on, the next time we open a project containing a linked image, Manifold will be able to use the copy of the image saved in the local disk cache and will not have to go through the process of downloading the image again.
The data cache will persist across sessions and projects so that once a tile is downloaded it will not have to be downloaded again.

The location of the cache folder is a TerraServer folder at whatever location is specified in the Data Cache item in Tools - Options - File Locations. The default value for the location is %MyDocuments%, so by default each user's files will be stored in the user's own cache folder.

When multiple users are working it might make sense to modify the default Data Cache setting for each user so the cache folder location is the same for all users and is on a shared resource available from all machines on the local network. For example, if there is a machine called Storage on our local network with a Cache folder on its C: drive, we might use a Data Cache setting of \Storage\Cache for all users. In that case, any tile files that are retrieved by one user will be available to all users so that tile files for a given view will never have to be fetched twice over the network. This assumes, of course, that the network is fast enough so that use of cache over the network will not be too slow.

It is also important when working with large TerraServer images that the location specified for the cache is on a disk drive with plenty of free space. Users sometimes will ask for download of TerraServer images without thinking about how large the resultant image may be and thus end up inadvertently requesting immensely large data sets.

The data cache folder may be moved to a different location so long as the path specified in the Data Cache option is adjusted to point to the new location.

Caution: Cache files persist forever and, in extreme cases, use up your free disk space unless they are manually deleted. See the recommended actions in the Managing Cache Files topic.

Notes

Not all imagery that can be fetched interactively from the TerraServer website is available for automatic download via the web service.

Even if we have a very fast Internet connection we must keep in mind that the connection to TerraServer may be much slower. Even though TerraServer uses very fast database technology, it is a free service that quite often is overloaded due to its great popularity. There may be times when images are very slow to load from TerraServer. To cancel a connection when using the Link TerraServer Data dialog, press the Escape key on the keyboard.

Using TerraServer Images in IMS Websites

When using TerraServer linked images within a Manifold Internet Map Server web page, keep in mind that linked images may take a long time to render as image tiles are fetched from the remote image server. For example, an IMS project that contains an image linked from TerraServer won't render that image on the website until the system fully downloads the entire required portion of the image from TerraServer.

The IMS page including the linked image can be no faster than the image server providing the image. If speed is of the essence, consider downloading the linked image into local storage, perhaps as a compressed image in ECW form.

The Manifold Image Server Interface (ISI)

Connection to TerraServer uses Image Server Interface (ISI) drivers built into Manifold to present TerraServer as a Manifold Image Server.

Manifold's Image Server Interface (ISI) is an open standard published by manifold.net to make it easy for third party developers to create drivers that connect image servers to Manifold System. Image servers that can work with Manifold System using an ISI driver are called Manifold Image Servers and can be automatically accessed as if they were a built-in part of Manifold System using the File - Link - Image dialog. See the Linked Images from Manifold Image Servers topic.

See Also
Selection in Images

Modifying Selections

See the introductory Selections topic for an introduction to selection in general and for a tutorial on basic selection methods that apply to all components.

This topic discusses a selection feature that is unique to images and surfaces. Certain selection functions make sense in images and surfaces but have no equivalent in other components. Such selection functions depend upon the component consisting of a continuous sea of pixels. In particular, when selecting pixels in images we often can refer to the selection as a region of particular shape. It is often highly desirable to be able to modify the shape of the region of selected pixels. We can do this with the Modify Selections commands or by Selection using Masks.

This topic uses images as examples. However, the techniques used apply to surfaces as well. Note that compressed images do not support selection.

Modify Selections

Selection modification commands are very useful when preparing images for use in vectorizing. In addition to general-purpose selection in images, they may be used to create "buffer zones" within images and other sorts of pseudo-analytical selections.

The Edit - Modify Selection menu provides four choices for modifying the shapes of regions of pixels that have been selected:

- **Border**: Take the current region of selection and convert it to region of selected pixels that are +/- the given width of pixels from the boundary of the current region.
- **Contract**: Shrink the region of selected pixels by the given number of pixels.
- **Expand**: Expand the region of selected pixels by the given number of pixels.
- **Smooth**: Create a smoother periphery of the selected region using the given parameters.

The Smooth function uses Threshold and Passes parameters. Each pixel on the edge of the selected region is evaluated by looking at its eight immediate neighbor pixels. If enough of the neighboring pixels are selected, then the central pixel will be selected. The threshold, a value from 1 to 4, states how many of the neighboring pixels must be selected before the central pixel is selected. The passes parameter states how many times this iterative process should be applied. The more it is applied, the smoother the periphery of the region of selected pixels will be.
Select touch was used in the image above to click on the sky. To eliminate the ragged edge of pixels at the periphery of the selection, we used Edit - Modify Selection - Smooth with the parameters shown below.

![Smooth dialog]

By increasing the number of passes we can make the periphery smoother as well as reduce the number of small "islands" of unselected pixels within the main body of the selection.

The selection modification commands are very useful when preparing images for use in vectorizing.

**Example: Using Modify Selection - Expand to create a Buffer Zone**

A "buffer zone" is simply a region that extends a certain distance beyond another region. For example, we might which to mark all pixels near the shoreline in an aerial photograph of waterfront development.

In this example we create a buffer zone of selected pixels that outline the upper part of the monument in our "bronze" sample image:

![Buffer Zone Image]

**Step 1: Select an upper region** - We do this by a combination of touch selection, Edit - Modify Selection - Smooth and then using select rectangle to subtract the lower region to leave a nice, smooth lower border. We use the Edit - Selections dialog to save this as a saved selection called "sky"
**Step 2: Select the inverse region** - Using Edit - Select Inverse to select the inverse region and then select rectangle with the mouse with Subtract from Selection mode to remove the bottom pixels. This leaves just the region in the upper part of the monument.

**Step 3: Expand the selected region** - Using Edit - Modify Selection - Expand we expand the region of selected pixels by 5 pixels.

This results in a region of selected pixels that extends outwards from the upper part of the monument.
Step 4: Use Intersect Selection Command from Selections Dialog - Within the View - Selections dialog we click on the sky saved selection. If the Preview button is checked, we will see a preview of the sky selection in blue. Note how it overlaps the expanded selection region created in the previous step.

Press the Intersect Selection command to intersect the sky selection with the expanded selected region. The selection now becomes the region of intersection, which is our desired buffer zone.

Using Modify Selection - Border

We can also create a buffer zone using Border selection modification. In this case, the "buffer zone" will extend both inward and outward from the edge of the previous selection region.

Suppose we start with a selection consisting of pixels in the upper part of the monument.
Using Edit - Modify Selections - Border with a setting of 5 pixels for the Width creates a zone of selected pixels extending both outward and inward from the previous periphery.

**Border Width**

The **Width** parameter is an approximate parameter in curved areas, since it is the number of pixel steps from the meandering line of pixels that is taken to be the center of the border region. The actual width of the border zone will be \([\text{width} \times 2 - 1]\) since the center line counts as one "width" amount.

Suppose we zoom into the image so we see the starting region of selected pixels at the head of the monument.
If we now run Border with a width of 1, a single line of pixels is selected at the periphery of the initial region. We can save this selection in the Selections dialog so that it may later be compared to other settings for width.

In the above illustration, instead of using a width of 1, we used a width of 3. We can see the previous setting of 1 in blue as a preview from the Selections dialog. Note how in vertical and horizontal areas where pixel stepping is clear we can see that the border region is five pixels wide because the "3" in the width parameter includes the center line pixel when counting out to either side. Thus, the border extends from the center pixel plus 2 pixels to either side.

**Touch Selection Tolerance Setting**

Touch selection will select all pixels whose color is within a given tolerance range of the pixel touched by the mouse. For RGB or RGBA images, tolerance is computed only on those channels that are made visible in the layers pane. For palette images, tolerance is computed based on the R, G and B values of the palette color as compared to other palette colors. Change the tolerance setting as desired in the Tool Properties pane to increase or restrict the range of pixel color values that are selected when using touch selection.
In the illustrations above we have increased tolerance from a low level to a much higher level and then clicked on the blue sky just above the head of the monument. As tolerance is increased a greater number of pixels are selected in colors that are more and more different from the blue hue of the touched pixel. At high tolerance settings pixels with colors very different from the touched pixel will be selected.

**Speed of Select Touch in Images**

By default, touch selection finds all pixels of similar color that are in a region connected to the touched pixel. Using `SHIFT`-touch selection finds all pixels of similar color whether or not they are in a contiguously connected region to the touched region. Determining whether or not pixels are within a contiguously connect region requires many spatial computations, so `SHIFT`-touch selection frequently will operate much faster than touch selection.
Selection using Masks

See the introductory Selections topic for an introduction to selection in general and for a tutorial on basic selection methods that apply to all components.

This topic discusses a selection feature that is unique to images and surfaces. Because images are made up of a continuous sea of pixels certain selection functions make sense in images but have no equivalent in drawings or tables. In particular, when selecting pixels in images we often can refer to the selection as a region of particular shape. It is often highly desirable to be able to modify the shape of the region of selected pixels. We can do this with the Modify Selections commands or through the use of Masks. Masks are used for purposes other than selection. See the Masks topic for other uses.

Masks

Masks are a way of using one image to make selections in another image. They are also a handy way of saving as many selections as we like for a particular image by saving the desired selection pattern to a file. A mask can be any grayscale image. There is nothing special about grayscale images used as masks. What makes a grayscale image a "mask" is simply how we use it.

We can create masks using any method used to create or edit grayscale images. Since masks are often used as a means of saving selections or other patterns from existing images Manifold provides an Edit - Save Mask/Channel command that makes it easy to create a mask using an existing image.

To Create a Selection Mask

1. Make a selection in the image.
2. Choose Edit - Save Mask/Channel.
3. In the Save Mask dialog, save the Selection to an image name.

This creates a new image that has white pixels in the region to be selected and black pixels elsewhere.

If we were to save the selection on the left as a mask the resultant mask image is shown on the right. Once saved, we can subsequently load the mask as a selection at any time we desire.
When we use \texttt{Edit - Load Mask/Channel} to make a selection in the image using a mask it is as if the mask were a cut-out in a sheet of cardboard through which we "spray" selection color onto the image, just like using a stencil.

\textbf{Masks are Images}

Masks are simply grayscale images like any other grayscale image. They may be opened, edited and otherwise changed as we desire. No selection will occur wherever there is a pure black (value 0) pixel. Pixels in any other color result in a selection being made at that spot. Masks used for selection are most often saved using just black and white colors so that the region to be selected is obvious.

We can create a new mask image at any time by simply inserting a new grayscale image into the project and then coloring it in black and white to show where the selection should occur. We can then use the "mask" image to make a selection. This allows us to specify regions that are to be selected using the full set of image editing tools within Manifold.

If we create the image at left above we can use it as a mask. Regions of white pixels will be selected. If we open the \texttt{bronze} sample image and use \texttt{Edit - Load Mask/Channel} to load the image as the Selection, we can select a region of pixels in the \texttt{bronze} image in the shape of the word "manifold."

We can create a selection mask by creating a grayscale image that has white pixels wherever we want a selection and black pixels where we do not want a selection. We can also create a selection mask by making a selection in an image and then using \texttt{Edit - Save Mask/Channel} to save it as a mask. We can combine the two methods, by first using \texttt{Edit - Save Mask/Channel} and then opening and editing the resultant mask.

\textbf{Matching Masks to Target Images}

Masks are usually the same size as the image on which they are to operate. If the mask is larger or smaller than the target image it will be used centered on the image and all pixels outside the mask will not be changed (that is, will not be selected).

Masks are often created to work with a particular image. For example, one might wish to create a selection mask that selected pixels only to one side of a riverbank in an aerial photograph. There are two frequently used methods to assure that the mask is the same size and otherwise matches positions in the target image:
• We can make a copy of the aerial photograph image and then edit that copy so that all the pixels not to be selected are black. We then convert the image copy to a grayscale image. If desired we can also use Threshold to force all non-black pixels into pure white.

• We can copy the aerial photo image and to position it as a partially transparent layer above the aerial photo image in a map. We can then “trace” the area desired using various editing tools to fill in the region not to be selected with black (or other convenient color) pixels. Using a map in this way makes it possible to create the selection mask by tracing over the target image as a guide where the guiding image will always be visible through the partially-transparent layer.

Invisible Pixels in Masks

Any invisible pixels in masks will not participate in whatever the mask is asked to do. If the mask is loaded as a selection, regions of invisible pixels will not be selected. If the mask is loaded as a channel, any parts of the mask containing invisible pixels will not alter the target image.

Additional Choices for Save Mask

When using **Edit - Save Mask/Channel** to save a mask, we have many choices in the dialog’s **Save** box in addition to saving the current selection as a mask. We can also save the mask using:

• The current selection.

• The region of invisible pixels.

• Any saved selection we have made for this image in the **Selections pane**.

• The R, G, or B channels in the image if it is an RGB image.

• The alpha channel in the image if it is an RGBA image.

See the Masks topic for a discussion on how the above options can be used to separate an image into different images for each channel.

Use with Surfaces

Masks can also be used with surfaces. Saving a mask from a surface results in an image that is georegistered to the surface and that retains any selection that was made in the surface. This is a handy way of saving complex selections as an alternative to using the **Selections pane**.

Editing Images

Editing Images

Editing images simply means changing the color/channel values of the pixels that make up the image and/or changing the size of the image. It's simple to say this, but there are numerous methods for doing so:

• Creating new images from drawings or other sources.

• Cutting, Copying and Pasting into the image.

• Drawing into the image “free hand” using image editing tools

• Using the Transform toolbar to change many pixels at once throughout the drawing.

• Using commands from the Image Menu such as resize to change the size or otherwise edit the image.

• Changing the image by projecting/georeferencing it.

• Altering images with scripts or other tools.

• Using Edit - Save/Load Mask/Channel to alter the image by channel (used to compose RGB images from multiple grayscale images).

• Edit the image using update queries that operate on the image's **virtual table** to directly change the colors of pixels or other characteristics. See the Virtual Tables for Images and Surfaces topic as well as the Queries and Images or Surfaces topic.

• Use commands
Not all editing effects will work with all types of images. Some effects will require RGB or RGBA images to function, since they require a continuous selection of many colors. Compressed images may not be edited.

**Resizing Images**

To resize an image by cutting away pixels or by adding pixels at the margin, use the Transform toolbar Crop, Crop Margin and Add Margin operators.

Changing the size of the image while retaining the visual appearance by inflating pixels or compressing them into fewer pixels requires resampling the image. Use the Image - Resize command to change the size of the image in such cases. This dialog allows a choice of methods to resample the image into a larger or smaller number of pixels.

**Cropping Images**

The Crop Margin on the transform toolbar for images crops the image down by the given number of pixels at the margin. The image is reduced in size and those pixels in the cropped margin are discarded.

The Crop command on the transform toolbar for images crops the image down to the minimum enclosing box for the selection. If there is no selection the Crop command has no effect.

The simplest way to crop images is to use **Select Box** to draw a rectangular selection box and to then launch crop. The image will be cropped to the size of the selection box. This is fast and works the same way that "crop" functions work in almost all graphics software.

In the image at left we’ve used **Select Box** to select the region enclosing Europe. We’ve used a **Border** selection style to show the selection. The image at right shows the result of the crop operation. The image has been reduced in size by the number of pixels cropped. It is shown on the checkerboard background within a larger image window.
Crop also works with irregularly shaped or discontinuous selections. Suppose we select pixels in South America as above.

Applying the crop operation using the Selection as the target will crop the image down to the size of the selection.

**Undo**

Image editing operations can be undone using Undo. The keyboard shortcut for Undo is **CTRL-Z**

**Zoom**

Images can be zoomed in and out. Use the View - Zoom to command to zoom to a specific level. The Native zoom choice in the Zoom To dialog will zoom the image so that one image pixel is exactly one screen pixel in size.
Selections and Editing

If a selection is made in an image any editing commands will be limited to the selected region only. This is a very important capability that is used to restrict the action of editing commands to desired parts of the image. It allows us to "paint" exactly up to a finite border without worrying that any paint will spill over into undesired portions of the image.

Suppose we use **Touch Select** to select all the pixels in our sample **bronze** image except those that make up the monument.

If we now switch the selection style to **Border** style so that the selection dots do not get in the way, we can draw a line using the **Insert Line** tool along the line indicated.

The line will be drawn only in those regions of selected pixels.
We can draw more lines in the image.

In all cases, only those pixels that are selected will respond to the drawing commands.

Here we will draw a circle using the Insert Circle tool.
Note that it paints over only those selected pixels.

All editing actions and not just drawing are limited to the selection. To create the above image we used Edit - Select Inverse to select just those pixels inside the monument and then we applied Threshold.

We can then follow up the Threshold command with a Gaussian Blur command, and then choose None for the selection style so that we can see the image without the red border defining the region of selected pixels. Note that the Gaussian Blur effect is sharp right up to the edge of the region of selected pixels.

See Also

Touch selection in images depends on the setting of Tolerance in the Tool Properties pane. A tolerance of 1 will select only those colors that are exactly the same as the color of the touched pixel. Increasing the tolerance value will allow colors that are less and less similar to the touched color to be considered the same.
Copy and Paste in Images
Copy and paste commands can transfer pixels between images. Pasted pixels will overwrite existing pixels in the same locations. When copying pixels from an image in one projection and pasting them into an image using a different projection, Manifold will automatically make any necessary coordinate system transformations.

Example

We select pixels from our bronze image and press CTRL-C to copy (or use Edit - Copy from the main menu).

We click on another image that is the same size as the bronze image and press CTRL-V to paste (or use Edit - Paste from the main menu).

The pasted pixels replace the original pixels. Because the destination image is the same size as the origin image the relative placement of the pixels is unchanged. The new pixels are selected.
We can deselect them to see the destination image as it now appears.

**Automatic Conversions**

When pasting from an image of one type (RGBa, RGB, palette or grayscale) to an image of another type, Manifold will automatically convert the pixels to the correct form for the destination image.

Suppose we have a palette image as seen above.

If we were to paste the same pixels copied in the previous example they would automatically be converted into the best match possible given the limited colors available in the palette image. Palette conversions when pasting will use default settings of the conversion algorithms.
If the pixels were pasted into a grayscale image they would be converted to grayscale. When pasted they would be selected.

Deselecting the pixels shows the grayscale conversion.

**Invisible Pixels are not Pasted**

Pasting invisible pixels has no effect upon pixels in the destination image.

Suppose we delete some pixels from the **bronze** sample image. Deleting pixels makes them invisible pixels.
We can then make a selection that includes both visible and invisible pixels. We can **Copy** the selection.

For this example we use a destination image in which pixels are colored in a black to white gradient created with the gradient tool.

When we **Paste** into this image the pixels will replace the original gradient pixels except in those spots where invisible pixels would be pasted.
Deselecting the pasted pixels shows that no changes were made to the original pixels in places occupied by invisible pixels.

**Pixel Locations**

Pixels will appear within the target image if their locations fall within the target image. When copying and pasting between two images of different size that are still in default, CAD-style coordinates, pixels will appear based on their X and Y distance from the lower left corner. To place pixels elsewhere, compose the images as layers in a map and then use **CTRL - grabber** to move the images as desired.

**Georegistration**

All pixels in Manifold images are georegistered to some point on Earth. If an image has not been imported from an image format that correctly captures projection location, it must be georegistered before it will make sense. Copying pixels from an incorrectly-georegistered image into a correctly-georegistered image will not move those pixels to the correct location. They will carry their already-defined (incorrect) location with them and thus will not appear within the destination image. See the Projections and Images topic.

**Copying Images and Pasting as Tables**

Images may be copied and pasted as tables. In that case, various attributes of the image will be available for pasting as **intrinsic fields** into the new table. See the Intrinsic Fields in Tables topic.
Masks

Masks are grayscale images that are used to control other images. Masks are used for two purposes:

- To make a selection in the shape shown by the mask.
- To alter the values in the R, G, B or Alpha channels of the target image using the values in the mask.

A mask can be any grayscale image. There is nothing special about grayscale images used as masks. What makes a grayscale image a "mask" is simply how we use it.

We can create masks using any method used to create or edit grayscale images. Since masks are often used as a means of saving selections or other patterns from existing images Manifold provides an Edit - Save Mask/Channel command that makes it easy to create a mask using an existing image.

Edit - Load Mask/Channel Options

Load Which part of the image is to be loaded. The selection, a channel, or invisible pixels.

From The source image to use for the mask. All grayscale images available will be listed together with their sizes.

Mode Specifies how to use the mask within the target image:
- Add - Add the intensity values of the mask pixels to the target.
- Invert - If the intensity of a mask pixel is greater than 128, invert the value of the target pixel.
- Maximum - Use the maximum of intensity of either the mask pixel or the target pixel.
- Minimum - Use the minimum of intensity of either the mask pixel or the target pixel.
- Replace - The default. Use the values of the mask pixels.
- Subtract - Subtract the intensity of the mask pixel from the target pixel.

Restrict to Selection If checked (the default), when a selection is present the mask will be applied only within the selected pixels.

Selection using Masks

Masks are a way of using one image to make selections in another image. They are also a handy way of saving as many selections as we like for a particular image by saving the desired selection pattern to a file.

To Create a Selection Mask

1. Make a selection in the image.
2. Choose Edit - Save Mask/Channel.
3. In the Save Mask dialog, save the Selection to an image name.

This creates a new image that has white pixels in the region to be selected and black pixels elsewhere.
If we were to save the selection on the left as a mask the resultant mask image is shown on the right. Once saved, we can subsequently load the mask as a selection at any time we desire.

When we use Edit - Load Mask/Channel to make a selection in the image using a mask it is as if the mask were a cut-out in a sheet of cardboard through which we "spray" selection color onto the image, just like using a stencil.

**Choices for Save Mask**

When using Edit - Save Mask/Channel to save a mask we have many choices in the dialog's Save box in addition to saving the current selection as a mask. We can also save the mask using:

- The current selection.
- The region of **invisible pixels**.
- Any saved selection we have made for this image in the Selections pane.
- The R, G, or B channels in the image if it is an RGB image.
- The alpha channel in the image if it is an RGBa image.

Masks created from any of the above can be used to subsequently control the selection; however, when saving the R, G, B or alpha channels the result is most normally used for as masks for channels or to separate the image into different grayscale images for each channel.

**Masks are Images**

Masks are simply grayscale images like any other grayscale image. They may be opened, edited and otherwise changed as we desire. When used as a selection mask, no selection will occur wherever there is a pure black (value 0) pixel. Pixels in any other color result in a selection being made at that spot. Masks that are used to make selections are usually "painted" in black and white colors even though the full range of grayscale colors may be used. This makes it easy to immediately see what is supposed to be selected and what is not.
Masks that are used to alter RGB or alpha channels are usually gray scale images of various kinds since a range of tones is sought for such purposes. The single value in each grayscale pixel in the range 0 to 255 is used for the value in the target channel.

We can create a new mask image at any time by simply inserting a new grayscale image into the project and then coloring it in black and white to show where the selection should occur. We can then use the “mask” image to make a selection or to control a channel. This allows us to specify regions that are to be selected using the full set of image editing tools within Manifold.

If we create the image at left above we can use it as a mask. Regions of white pixels will be selected. If we open the bronze sample image and use Edit - Load Mask/Channel to load the image as the Selection, we can select a region of pixels in the bronze image in the shape of the word “manifold.”

We can create a selection mask by creating a grayscale image that has white pixels wherever we want a selection and black pixels where we do not want a selection. We can also create a selection mask by making a selection in an image and then using Edit - Save Mask/Channel to save it as a mask. We can combine the two methods, by first using Edit - Save Mask/Channel and then opening and editing the resultant mask.

**Masks and Channels**

Before continuing with this topic, please read the Images and Channels topic and the RGBA Pixel Transparency topic if you have not yet done so.

In the example above, we used the Edit - Load Mask/Channel to load the Selection using a mask as a guide. We can also use the Load Mask command to load a channel using the contents of a mask as a guide. Using a mask we can load any of the following aspects of an image:

- The Selection.
- Make the pixels in a given region invisible pixels.
- The R, G, or B channels in the image if it is an RGB image.
- The alpha channel in the image if it is an RGBA image.
Suppose we saved the selection shown at the beginning of this topic as the above mask image, which we called mask1.

We can load it into the green channel for the bronze image with the Edit - Load Mask/Channel dialog by choosing Channel: green in the Load box.

This deletes all the previous values in the green channel of the bronze image and replaces it with the "all on" or "all off" values for green the mask specifies.

It is as though we spray-painted green color into the green channel using the mask as a stencil.
When combined with the unmodified red and blue channels in the **bronze** image the result is the above.

Again, it is as if we deleted all the green from the image and then sprayed pure green color into the image through the mask as a stencil. Note how the clouds are no longer white (because for the most part they no longer have any green color in them). Instead, the clouds are mostly magenta, which is the color that results from mixing bright red with bright blue.

Note that masks don't have to be used only with those images that were used to create them. We can load the sample mask above into the red channel of our sample castle image if we like.
We can use completely new masks that we create on our own with images. Here we have loaded the "manifold" mask shown earlier in this topic as the red channel in the sample bronze image. Note how the clouds now appear cyan, which is the color obtained with no red but very bright blue and bright green.

The illustrations above show simple, black and white masks. Grayscale images can also be used as masks to continuously vary the R, G, B or alpha channel in images.

**Using Masks to Specify Alpha Transparency**

Masks may be loaded into the alpha channel of images to control transparency.

Black regions in the mask mean zero transparency and white regions mean full transparency. Gray tones in between cause partial transparency.
Using the sample mask we created at the beginning of the topic results in the above pattern of transparency where the image is either fully transparent or not at all transparent.

If we take our sample mask, convert it to RGB, apply Gaussian Blur to blur it, and then convert it back to grayscale we can use it as a mask to create a transitional blend of transparency between fully transparent and fully opaque regions.

The image above shows what happens when we load the blurred mask into the alpha channel. Regions where the mask was completely black have no transparency and are fully opaque. Regions where the mask was completely white are completely opaque. The feathered edges in between where the mask has a blurred gradient of gray tones from black to white now have a blurred gradient of transparency from fully transparent to fully opaque.

One of the most frequent uses of Edit - Load Mask/Channel is to load a grayscale gradient to cause a continuously changing transparency gradient.
We can use the grayscale image of a gradient from black to white shown above as a mask for the alpha channel in our sample bronze image.

The result is a continuous variation of transparency from fully opaque (the white regions of the gradient) to full transparency (the black regions of the gradient).

**Using Masks to Specify Invisible Pixels**

We can use a mask to specify any regions of invisible pixels.
Suppose we would like the yellow star shown above to shine through our sample bronze image. To do this we will cut out a region of invisible pixels in the bronze image in the shape of the star.

We begin by creating a mask from the star image. To do this we first used Threshold to make the star entirely black and everything else white and then we used Invert to get the above image. This was converted into a grayscale image using Convert to.

We open the bronze image and then use Edit - Load Mask/Channel to load the invisible pixels using the star mask image created above.

The result is a bronze image with opaque pixels in black areas of the mask and invisible pixels in the white regions. The cutout region of invisible pixels is perfectly aligned with the edges of the yellow star because the yellow star was itself used to create the mask.
When we combine the two images in a map with the star image underneath the bronze image the yellow star shines through the bronze image.

**Applying Masks only to Selected Pixels**

If we check the **Restrict to Selection** box in the **Load Mask / Channel** dialog the mask will be applied only to selected pixels if a selection exists.

Let's use the image above as a sample image. We created this image using Image Editing Tools employing a variety of brushes, color and opacity settings. A region in the lower left corner has been selected.

The image above shows a simple mask created using a solid black image to which the Noise effect has been applied. This created an image with speckles of white on a black background.
If we load the Blue channel in the image with the mask using **Replace** mode (the default), the mask will be loaded only into those pixels that are selected if the **Restrict to Selection** box is checked.

This works no matter what pixels are selected. In the illustration above we have selected five circular regions of pixels.

If we load the Green channel with the mask with **Restrict to Selection** checked the mask will be applied only within the selected regions.

**Mode Choices**

The **Mode** option in the **Load Mask/Channel** dialog allows us to choose how we would like the mask to be used in the target image.

The default mode is **Replace**. Loading a mask with **Replace** mode means the values in the mask will replace the values for the channel loaded. Note for example in the illustration above how the yellow color in the central
circular region has turned red. That happened because the color yellow is a combination of red and green. When loading a mask into the Green channel that has white dots on a black background there is no green value in the black portions of the mask. Where the black portions of the mask replaced green channel pixels no green color is left and so only the red channel is left in what were originally yellow pixels.

Suppose we make a different selection in the original sample image.

If we load the mask into the Green channel using **Add** mode, note that the yellow pixels are unaffected where the mask is black. That is because black results in adding zero to the green channel intensity. Where the mask has white dots, the yellow is a brighter green-yellow because adding 255 to the green channel forces it to be full green. In the blue pixels of the target image the result is to create bright cyan (blue plus max green) in the spots where the mask has white pixels.

If we use **Subtract** mode, the values in the mask will be subtracted from the Green channel. There is no change in black parts of the mask since subtracting zero results in no change. In the yellow pixels of the target image, subtracting 255 in those spots where the mask is white results in total absence of Green and thus results in red pixels. Subtracting 255 from the Green channel in pure blue parts of the image has no effect since there is no Green there to begin with. Subtracting 255 from the green parts of the image results in black pixels.
A GIS Example Using Mode

The Mode option to combine a mask by adding it, subtracting it or otherwise combining it in a sophisticated way with an existing channel provides a powerful means to combine multi-spectral images into powerful and highly informative false color images. This example is copied from the Combining Channels into Images topic.

Using a straight, unprocessed combination of the Red, Green and Blue bands from the Landsat 7 Enhanced Thematic Mapper instrument (bands 3, 2 and 1) as the Red, Green and Blue channels in an image results in the muddy and uninformative image above.

We can enhance the image by using a Band 4 image as a mask and using Add mode to add its intensity values to the Green channel of the image. Band 4 is a reflective infrared band that is highly responsive to vegetation. By using it to "kick" the Green channel up we enhance the green element normally associated with vegetation.

We can further enhance the image by using a Band 7 image as a mask and Add mode to add it to the Red channel of the image. Band 7 is a lower mid-infrared band that is responsive to soil types. This lets us see not only vegetation but also soils.

See the Combining Channels into Images topic for additional examples of Landsat 7 image processing.

Matching Masks to Target Images

Masks are usually the same size as the image with which they are used. If the mask is larger or smaller than the target image it will be applied centered on the image and all pixels outside the mask will not be changed.

Masks are often created to work with a particular image. For example, one might wish to create a selection mask that selected pixels only to one side of a riverbank in an aerial photograph. There are three frequently used methods to assure that the mask is the same size and otherwise matches positions in the target image:
• We can make a copy of the aerial photograph image and then edit that copy so that all the pixels not to be selected are black. We then convert the image copy to a grayscale image. If desired we can also use Threshold to force all non-black pixels into pure white.

• We can copy the aerial photo image and to position it as a partially transparent layer above the aerial photo image in a map. We can then "trace" the area desired using various editing tools to fill in the region not to be selected with black pixels. Using a map in this way makes it possible to create the selection mask by tracing over the target image as a guide where the guiding image will always be visible through the partially-transparent layer.

• Use the Match command within a map to automatically "cookie-cut" one image with another and to adjust the pixel density and coordinate systems to match.

**Invisible Pixels in Masks**

Any invisible pixels in masks will not participate in whatever the mask is asked to do. If the mask is loaded as a selection, regions of invisible pixels will not be selected. If the mask is loaded as a channel, any parts of the mask containing invisible pixels will not alter the target image.

For example, when using the mask at left above to select pixels in the image at right above the region of invisible pixels will not result in a selection.

**Saving Channels as Separate Images**

At any time we can save the separate channels of an RGB or RGBA image as different grayscale images. To do so, we use the Edit - Save Mask/Channel command with the channel to be saved specified in the dialog's Save box.

See the Separating Images by Channels for a discussion on how Edit - Save Mask/Channel can be used to separate an image into different images for each channel.

**Nomenclature**

Transparency and opacity are two terms that mean the same concept viewed from different directions. When something is completely opaque it is not at all transparent. When something is perfectly transparent it may be said to have zero percent opacity.

Which word is used depends on the discussion. When imagining layers stacked up above each other like transparent sheets it is conceptually clearer to use the word transparency. When discussing a specific percentage of light transmission to be applied via a slider bar in a dialog most applications use the word opacity.

The convention in the graphics arts editing software industry is to adjust layer transparency with controls that set a number from 0% to 100% opacity, so that an image with 100% will be fully opaque and not allow any view of an image underneath it. Manifold follows this convention. This convention persists in the graphics arts industry even though the technical implementation of transparency effects is done using an alpha channel within RGBA images where the higher the value of the alpha channel (from 0 to 255) the higher the transparency.
One therefore encounters the slight conceptual dissonance of increasing opacity with higher numbers (up to 100%) in dialogs and other user interfaces while the internal data sets use numbers (alpha channel values) in which opacity decreases with higher numbers. Since we rarely set alpha values by hand this is not so bad. Alpha values are normally set using various tools, such as erasers, or masks. In the case of masks, the darker the mask region the lower the alpha value is and thus the higher the opacity. From a casual conceptual view this is very acceptable because it leads to an effect where black regions of masks cause full opacity and white regions of masks cause full transparency. Since we are used to thinking of "white space" as being transparent this works well as a natural mnemonic for the effects of masks.

See Also:

Images and Channels
RGBa Pixel Transparency
Painting within Channels
Separating Images by Channels
Combining Channels into Images
View - Panes - Layers
Match
Painting within Channels

Editing effects within Manifold are applied only to those channels that are checked ON in the View - Panes - Layers pane. Painting commands will also be applied only to those channels that are checked ON. If you have not done so, please read the Images and Channels topic and the RGBA Pixel Transparency topic before proceeding.

Let's begin by filling an image with black color so items we draw will be clearly visible. If we draw a triangle using white color it will appear white with all three R, G, B channels ON. White paint is a color that has maximum values in Red, Green and Blue.

If we check OFF the Red and Blue channels and check ON the Green channel the triangle appears green because we only see the green values of the pixels.

Checking the Red channel ON and the other channels OFF shows only the red color values in the white triangle.
If we continue to use white as the color of our painting tool and add three dots with only the Red channel ON, they will appear in red color.

If we switch ON all three RGB channels we see the white triangle appears in white again but the dots drawn using white paint color are still red. Because the other two channels were OFF when the dots were drawn their pixels were created using only the Red channel values.

We can now click the Red channel OFF and the Blue and Green channels ON. The triangle appears cyan in color because cyan is the color obtained from an even mix of Blue and Green with no Red. Note that the three small dots are not visible because their pixels have no color values in the Blue or Green channels. If we draw a large dot using white paint it will be created in cyan color because the blue and green values from the white paint color will be written into the pixels using the active Blue and Green channels.
We now click OFF the Green channel so that only the Blue channel is ON and then we create a very large dot using white paint color. It appears blue because only the blue values from the white paint are written into the active Blue channel. The triangle appears blue and the other dot appears blue as well. We see only the blue part of the (blue + green) cyan color for the cyan dot. We do not see the three small dots because they do not have any blue values in their pixels.

If we click ON all three channels we can see the image as it has been created. All pixels were painted using white color. Only the white triangle was painted with all three channels ON so only it is white. The red dots were painted when only the Red channel was ON so they only have red color values. The large blue dot was painted when only the Blue channel was ON so the pixels in the large blue dot only have blue values. The pixels in the cyan dot were painted when both the Blue and the Green channels were ON so they have both blue and green values.

**Painting into the Alpha Channel**

When painting into an image's R, G and B channels the color values for each channel come from the separate R, G and B values of the paint color being used. When painting into the Alpha channel the value for each pixel comes from the **intensity** of the paint. Although any color paint may be used it is usually easiest to reckon the intensity by using different shades of gray. Because true gray colors have equal mixes of R, G and B color values the intensity of any shade of gray is simply the overall appearance of lightness or darkness of the gray color.

Let's take a moment to convert our RGB image into an RGBA image using the Image - Convert To command. We will now see an Alpha channel in the Layers pane in addition to the Red, Green and Blue channels. If we click the Alpha channel ON and all the other channels OFF we can paint into the Alpha channel without altering the other
channels. We can paint a series of dots into the Alpha channel using a lighter shade of gray color for each dot. The result is a series of dots that have increasing amounts of transparency (the lighter the gray color, the greater its intensity and thus the greater the transparency).

If we click ON all four channels we can see how the differing transparency levels affect the image. In particular, we can take a close look at the blue dot, the cyan dot and the uppermost red dot to see how parts of the dots are now semi-transparent as set by the alpha transparency.

If we layer the above image in a map above our sample bronze image we can see how the transparency applied by painting different shades of gray dots into the Alpha channel now allows more or less of the bronze image to be seen through the upper image. Note how the blue, cyan and uppermost red dots are partially transparent in regions where the gray dots were painted into the Alpha channel.

See Also:

Images and Channels
RGBa Pixel Transparency
Masks
Separating Images by Channels
Combining Channels into Images
View - Panes - Layers
Separating Images by Channels

We can separate RGB images into three grayscale images that contain the color values for each R, G, and B channel by saving each channel as a mask. We can separate RGBa images into four grayscale images for each R, G, B, and alpha channel. To do so, we use the Edit - Save Mask/Channel dialog. See the Masks topic for detailed discussion on masks.

Separating an RGB Image into Separate, Per-Channel Grayscale Images

1. Open the RGB image to be separated.
2. Use Edit - Save Mask/Channel to save the red channel into a grayscale image.
3. Use Edit - Save Mask/Channel to save the green channel into a grayscale image.
4. Use Edit - Save Mask/Channel to save the blue channel into a grayscale image.

Example

We can use the Layers pane to see the separate R, G, and B channels in an image. We can save each of these channels using Edit - Save Mask/Channel to save the channel as a grayscale image.

The image on the left shows the red channel. The image on the right shows the grayscale image created as a mask from the red channel. Note that the sky is dark in the grayscale image because pixels in the blue sky have low values in their red channel.
We can create the image on the right from the red channel of the bronze image by choosing Channel: red in the Save: box of the Edit - Save Mask/Channel dialog.

Likewise, we can create a grayscale image for the green channel. Note that the bronze monument has bright tones in it because of the green color of the bronze monument.

Finally, we can create a grayscale image for the blue channel. Note how the sky is the lightest in this grayscale image because of the large amount of blue in it.

We can apply editing effects to the separate images and then recombine them back into an RGB image using the Edit - Load Mask/Channel dialog. See the Combining Channels into Images topic for an example.

Note that we do not need to split up an RGB image into separate grayscale images just to apply effects on a per-channel basis. Editing commands apply only to those channels that are visible. If we wish to apply commands to only one channel at a time we can uncheck the other two channels in the Layers pane and they will remain unaffected.
Combining Channels into Images

We can create RGB images by loading three different grayscale images into the R, G and B channels of the new RGB image using Edit - Load Mask/Channel. The most frequent application is creating a false color RGB image from multi-channel satellite sensor data. See also the Masks topic for more information on using images as masks.

Combining Per-Channel Grayscale Images into One RGB Image

1. Create an image of the desired size. This can be done by simply copying one of the channel images and renaming it.
2. Use Edit - Load Mask/Channel to load the red channel with the desired grayscale image.
3. Use Edit - Load Mask/Channel to load the green channel with the desired grayscale image.
4. Use Edit - Load Mask/Channel to load the blue channel with the desired grayscale image.

Example

The classic example of such work is creating false color RGB images using infrared bands from the Landsat satellite's thematic mapper sensor.

Using its Enhanced Thematic Mapper instrument the Landsat 7 satellite delivers images in seven standard Landsat bands plus a higher resolution panchromatic band from green color to near infrared. All the standard seven bands provide 30 meter resolution except the thermal infrared band, which delivers 150 meter resolution. The panchromatic band achieves 15 meter resolution.

<table>
<thead>
<tr>
<th>Band</th>
<th>Wavelength (micrometers)</th>
<th>Description</th>
<th>Resolution (meters)</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.45 to .52</td>
<td>Blue</td>
<td>30</td>
<td>Discriminates between soil and vegetation. Map coastal water areas.</td>
</tr>
<tr>
<td>2</td>
<td>.52 to .60</td>
<td>Green</td>
<td>30</td>
<td>Shows healthy vegetation.</td>
</tr>
<tr>
<td>3</td>
<td>.63 to .69</td>
<td>Red</td>
<td>30</td>
<td>Discriminates between plant species. Useful in geology.</td>
</tr>
<tr>
<td>4</td>
<td>.76 to .90</td>
<td>Reflective infrared</td>
<td>30</td>
<td>Crop identification. Shows vegetation well.</td>
</tr>
<tr>
<td>5</td>
<td>1.55 to 1.75</td>
<td>Mid-infrared</td>
<td>30</td>
<td>Shows water content of plants well. Detects drought.</td>
</tr>
<tr>
<td>6</td>
<td>10.4 to 12.5</td>
<td>Thermal infrared</td>
<td>150</td>
<td>Low resolution band. Shows thermal pollution and geothermal sites.</td>
</tr>
<tr>
<td>7</td>
<td>2.08 to 2.35</td>
<td>Mid-infrared</td>
<td>30</td>
<td>Geologic uses. See changes in soil types.</td>
</tr>
<tr>
<td>PAN</td>
<td>0.50 to 0.90</td>
<td>Green to infrared</td>
<td>15</td>
<td>Panchromatic (gray scale). Highest resolution. Co-registered with other bands.</td>
</tr>
</tbody>
</table>

We will take a sample Landsat 7 image from bands 4, 5 and 7, the most useful infrared bands.
Band 4 we will use as the green channel since it is most sensitive to the mass of vegetation at a particular location.

Band 5 we will use as the red channel since it is brighter and has better contrast than the band 7 image. This should provide a good distinction from the green channel.

Band 7 we will use as our blue channel. Since this is the darkest image, the resulting composite image should be mostly greens, reds and some yellows.

The resulting RGB image is a typical "false color" rendition of colors invisible to human eyes (from the infrared part of the spectrum in this case) translated into visible colors. We created this image by loading the above bands into the R, G and B channels using Edit - Load Mask/Channel. The image brings out areas of large plant biomass and also shows regions with contrasting water content of plants.
If desired, we can alter the various grayscale images before combining them into the false color image. This image takes band 7 and inverts it using an inversion center that's below the default so that the image does not become too bright.

If we load this new, inverted band 7 grayscale image into the blue channel we get an entirely different look to the image.

**Using the Mode Option**

The *Mode* option to combine a mask by adding it, subtracting it or otherwise combining it in a sophisticated way with an existing channel provides a powerful means to combine multi-spectral images into powerful and highly informative false color images.

Using a straight, unprocessed combination of the Red, Green and Blue bands from the Landsat 7 Enhanced Thematic Mapper instrument (bands 3, 2 and 1) as the Red, Green and Blue channels in an image results in the muddy and uninformative image above.
We can enhance the image by using a Band 4 image as a mask and using Add mode to add its intensity values to the Green channel of the image. Band 4 is a reflective infrared band that is highly responsive to vegetation. By using it to "kick" the Green channel up we enhance the green element normally associated with vegetation.

We can further enhance the image by using a Band 7 image as a mask and Add mode to add it to the Red channel of the image. Band 7 is a lower mid-infrared band that is responsive to soil types. This lets us see not only vegetation but also soils.

See Also

See the Import Landsat Files and Create Composite RGB Image example for a detailed example using Landsat bands. The example shows how to import typical Landsat data, how to create a composite RGB image and how to use image enhancement techniques to improve image readability. Finally, the example shows how to combine the 15 meter resolution panchromatic band with an RGB image created from 30 meter bands to improve overall resolution.
Composing Complex Images in Layers
Using layers in a map we can create complex images by building up images within layers.

This image is composed of eight layers, beginning with our sample bronze image.

We used the bronze image to make selections of the sky, clouds, the monument and the pedestal and to save them in the Selections pane. The illustration above shows the clouds selection, which was built up using Touch - Select and Add to Selection mode. These selections were saved as masks for future use. The selections were used to copy pixels out of the initial bronze image and to paste them as new images.

The first layer on top of the base image is an image created by taking an image with just the white clouds, colorizing them and then applying a Gaussian Blur to make a halo. This layer was moved up a few pixels relative to the rest of the image.
The clouds image by itself is transparent except for the clouds.

When it is placed in a layer over top of the halo, the clouds cover the halo except where it extends above them and in the gaps between clouds where the legs of the monument are located. These gaps will be covered by layers above.

The next layer is the drop shadow for the monument, which has been moved a few pixels to the right and down.
This provides a credible shadow effect when the monument itself is added in a layer above.

We now will add images in layers above for drop shadows for the text as well as the pixels for the text.

Like all the drop shadows, this shadow was created from the text image using Gaussian Blur.
The "Manifold" text line was created by making a screen shot of the Manifold text and pasting it into an image. We could then use **SHIFT - Select Touch** to select all the red pixels that made up the letters, choose **Edit - Select Inverse** to select all the pixels except the red letters and then choose **Edit - Delete** to make everything except the text invisible pixels.

One nuance of how we created the image is that the drop shadow for the monument covers the upper part of the pedestal, as can be seen in the illustration at left above. To remedy this, we selected the pedestal pixels, copied them and then pasted them as a new image layer at the top of the layer stack. As seen in the illustration at right above, this covers up the drop shadow that we do not want in front of the pedestal.

Images composed of many layers using a map are very easy to change. We created the above image by moving the Manifold text and drop shadow image layers below the monument and monument drop shadow layers in the map's layer stack. We then moved both the Manifold text image and the text drop shadow image up relative to the other images.

**Combining Tiled Images into One Image**

At times we may wish to mosaic or tile several different images into a single image. Perhaps, for example, we have several aerial photographs of a region that together could form a single image. There are two cases to consider. The first case is where we have georegistered images that can be seen in their correct geographic position within a map. The second case is non-georegistered images that when seen together as layers in a map will appear all on top of each other.

**Combining images into a single image (georegistered images)**
1. Assemble all the images in a map, each in its separate layer.
2. Make any adjustments in contrast, brightness or other visual characteristics to provide an even visual appearance between adjacent images.
3. Create a single image from the tiled map stack by using the Tools - Make Image command.

**Combining images into a single image (non-georegistered images)**

1. Assemble all the images in a map, each in its separate layer.
2. The images (see Repositioning Layers ) to their desired locations.
3. Make any adjustments in contrast, brightness or other visual characteristics to provide an even visual appearance between adjacent images.
4. Create a single image from the tiled map stack by using the Tools - Make Image command.

Manifold is different than programs like PhotoShop, where one has a Merge Layers command. The reason is that all images in Photoshop are the same pixel size so one can merge layers by simply overlapping pixels.

In Manifold, images can have all sorts of different pixel sizes. a 400 x 600 pixel image could be georegistered to cover the entire Earth or it could be a 1200 DPI image. For that matter, an 800 x 1000 pixel image could cover the same surface area as the 400 x 600 image. We could see these two images tiled next to each other in a map even though the one image has "smaller" pixels than the other one. To merge them into one image, we have to say what pixel size we want that final image to have. This is stated by implication when we specify the size of the combined image in the Save Image to dialog.

**Tech Tip**

Sometimes mosaic images overlap each other but do not match well at their edges due to photographic distortion or other effects. In such cases we can achieve a better visual blending of the images if we convert the "upper" image to an RGBA image and then use pixel transparency effects to "feather" the edge of the image from full opacity to full transparency. This can provide a blending effect that gives a better visual match to the adjacent image than a hard edge. See the RGBA Pixel Transparency and Create a Circular Feathered Image topics for relevant techniques.
Editing Palettes
Palettes may be edited interactively to edit images by altering the colors that are in the palette.

Double click on a palette component to open it in a palette window. Make any changes by selecting one or more color wells in the palette and then changing the color. Wells may be selected by clicking onto a color well for an individual color to select it, or by selecting more than one color well any standard Manifold selection method. Any changes made to a palette will immediately be applied to images that use that palette.

As is the case with other component windows in Manifold, when no mouse command mode is in force the mouse operates in "smart mode": clicking on a color well will select it. Double-clicking into a selected color well will change the color.

Example

Let's begin with our sample bronze image converted to a palette image with a limited number of colors.

We can open the palette in a palette window and click on one of the colors in the last row of color wells to select it. The palette color selected is the lightest color, which is used in the palette image for most of the pixels in the clouds behind the monument.

Double clicking into this color well allows us to change the color if desired, say, to bright green.

The result will immediately appear in the image. Note that all pixels using this palette color have changed to bright green.
We can change another color in the palette to bright yellow.

All pixels in the image using that color will be instantly displayed in bright yellow.

Image Editing Tools

Image Editing Tools

Image editing in Manifold uses several controls at once to achieve exactly the desired effect. To edit images we can choose:

- Image menu commands. These commands perform global changes on the image or the selected area, such as changing hue or contrast. See the Image Menu topic.
- Transform toolbar commands. These are simple commands or one-line versions of Image menu commands that change the entire image or just the selected area. See the Transform Toolbar - Images topic.
- Editing tools. These are found on the Tools toolbar for image and provide a variety of free hand and assisted editing capabilities.

Editing Tools

When we paint into an image, we choose a tool and then set any options for that tool.

- Tools are selected from editing toolbars.
- The specific action of tools is set in the Format Toolbar or in the Tool Properties pane.

The Tools Properties pane should be kept open when editing images so that parameters can be changed as desired.

To Edit an Image:

1. Choose a tool.
2. Adjust tool properties (such as color to be used) in the Tool Properties pane.
3. Use the tool in the image.

Tools Toolbar for Images
The Tools toolbar for images hosts the image editing tools. Image editing tools are for the most part analogous to drawing editing tools. There are a few modifications specific to the pixel nature of images.

- **Paint Areas** - A mode button: the selected tool will create filled regions of pixels.
- **Paint Lines** - A mode button: the selected tool will create lines.
- **Paint Points** - A mode button: the selected tool will create points where the mouse cursor changes direction on clicks.
- **Paint Area** - Paint pixels within the region clicked. Creates lines or points if shape modes so specify.
- **Paint Freeform** - Paint pixels within region about which the mouse is clicked and dragged.
- **Paint Line** - Paint a line defined by straight-line segments between clicks.
- **Paint Box** - Paint a rectangular box using a mouse click and a drag.
- **Paint Box on Center** - Paint a rectangular box centered on initial mouse click.
- **Paint Circle** - Paint a circle using a mouse click and a drag.
- **Paint Circle on Center** - Paint a circle centered on initial mouse click.
- **Paint Ellipse** - Paint an ellipse using a mouse click and a drag.
- **Paint Ellipse on Center** - Paint an ellipse centered on initial mouse click.
- **Brush** - Paint a dot with one click or a continuous line with a click and a drag. Paints unsmoothed freeforms even if smoothing is turned on. Very useful for single pixel editing (click on a pixel to color it) at high zooms.
- **Paint Bucket** - Click to fill a contiguous region of similarly colored pixels with foreground color. Fills to a given tolerance value of color. SHIFT click to pour into the given color in all regions of the image within the given tolerance, even if not contiguous. Use this as a "replace color" operation.
- **CTRL** click to replace colors within the threshold while retaining the same intensity.
- **ALT** click to paint color without regard to threshold. Used to fill the entire selection or the entire image (enabled if a snap mode is not set).
- **CTRL** and **SHIFT** or **CTRL** and **ALT** may be combined.
- **Gradient** - Click, drag and release to create a color gradient from foreground color to background color between the click point and ending at the release point.
- **Pick Color** - Click to pick the color at that location as the foreground color. Shift click to pick the color at that location as the background color.

**Regular and On-Center Commands**
Like selection tools, painting tools occur in both on-center and regular versions. We can see the difference between the regular tool and the on-center tool by painting a circle.

Using Paint Circle to click near the center of the monument above and then dragging to the spot shown will open a preview circle as shown. Releasing the mouse button paints within the indicated preview circle. With regular commands, the mouse shows the desired diameter of the circle. The Paint Circle command appears by default on the Tools toolbar for images.

We can use Paint Circle on Center to click at the same beginning location and then drag to the same ending location. Note that the preview circle is larger than the previous circle. With on-center versions of commands the initial click is the center of the circle and the mouse motion shows the desired radius of the circle. Use this command to draw circles centered on a given spot.

Frequently used controls

- **Tools**: Pick one of the editing tools from the tools toolbar. The choice of tool sets overall behavior, such as whether each click results in a point or whether we create a rectangle by dragging open a mouse box.

- **Brushes**: Manifold has many brush styles that may be selected in the format toolbar. Combinations of different brushes with different tools will result in different painting effects.

- **Color**: Most tools use only foreground color, as set in the format toolbar.

- **Size**: The Size parameter in the format toolbar specifies the size of points and the width of lines in pixels. It is applied to enlarge or decrease the size of brush used. Some brushes require a
Images

A technical parameter that defines the relative height of the density curve used for tools that paint pixels using a curve to define the amount of paint applied. Similar in visual effect to the "hardness" tool parameter in PhotoShop.

Tolerance Used with selection, paint bucket fills, etc. Specifies how close a color must be to the color specified before it is considered the same color. Increasing the tolerance value will allow colors that are less and less similar to the touched color to be considered the same.

Brushes

Brushes specify different patterns in which paint is applied. They have a distinct visual effect when painting points or lines. Areas are always filled with solid color when painting filled regions such as a rectangular box.

Varying tools, colors brushes and size can result in dramatically different visual appearance. The above illustration shows different brushes in different sizes applied with different colors. Reduced to 256 colors to fit into this Help documentation it is but a pale rendering of the original, vivid True Color image. Create your own to see what we mean.
When painting lines brushes will have the same effect as if they were dragged in the direction the line is painted. The illustration shows lines drawn using a fuzzy round brush in different colors and sizes.

**Size**

Changing the size parameter changes the size of points and the thickness of lines. It has no effect when creating filled regions of pixels using area commands.

The illustration above shows a brush used to make points with several different size settings, a box command using two different size settings for the line, and a single pixel sized curly line drawn freehand.

**Paint Mode**

Just like drawings, editing tools in images can automatically create their desired effects as combinations of points, lines or areas. In images, of course, these are not objects but just point-shaped, line-shaped or area-shaped regions of pixels. The main use of paint modes is to specify whether shapes such as boxes, circles or ellipses are created using outlines or as filled regions.

Graphics editors will often have separate commands for creating an outlined rectangle or a filled rectangle. Other graphics editors always create an outlined rectangle and expect that the user will use a paint bucket command to fill the rectangle if a filled rectangle is desired.

Manifold specifies how a rectangle should be created by whether the **Paint Lines** or **Paint Areas** button is pushed in. Pushing in the **Paint Areas** button creates a filled region of pixels with sharp edges like the box on the right in the illustration above. Pushing in the **Paint Lines** button only will create the region as an outline using whatever brush is specified in **Tool Properties** to create the line. A box created using a square brush illustrated at left in the illustration above.
Pushing in both the **Paint Areas** and the **Paint Lines** button will result in a filled region that is outlined by a line using the given brush style. The illustration above shows various combinations of the **Paint Areas**, **Paint Lines** and **Paint Points** shape modes using a fuzzy round brush set to a size of 20 pixels.

Tools will paint using whatever brush has been selected. The box at left in the illustration above was painted using a round brush for the line. The box at right was painted using a square brush for the line. Note that the square brush results in sharp corners when a **Size** greater than one is used and a round brush results in round corners.

Note that using a **size** larger than 1 pixel together with the **Line** button will result in a region that is larger that the centerline indicated by the mouse since lines are drawn on centerline. A line fatter that one pixel will extend out from the centerline drawn.

**Automatic Paint Modes**

By default, the **Tools - Options** parameter **Automatic Paint Modes** is checked ON. Automatic Paint Modes will automatically switch paint modes whenever a new image editing tool is chosen to provide the most likely paint mode for that tool. For example, when Automatic Paint Modes is ON, pressing **Paint Line** will automatically push in the **Paint Lines** mode button and push out the **Paint Areas** and **Paint Points** buttons. This default choice may be overridden by choosing whatever combination of paint modes is desired after choosing the tool.

For example, when Automatic Paint Modes is ON the **Paint Areas** and **Paint Box**, **Circle**, and **Ellipse** commands will paint areas of pixels only. To paint lines and points as well simply push IN the **Paint Lines** or **Paint Points** buttons after choosing these tools.

**Density**

Some brushes use a mathematical curve to define the amount of paint sprayed into various locations of the brush. The **Density** factory specifies the proportional difference between the maximum and minimum amount of ink sprayed into different parts of the brush as defined by the curve. In effect, it is either flattening or accentuating the curve.

The effect of increasing this factor is to give a greater appearance of paint density, hence the name of the parameter. The illustration above shows a typical curve-defined brush used to create a “point” with the density factor used for each example printed below in white. At low density the difference between the different regions of the brush are greatest and so it appears as a less dense brush. With high density there is less difference between paint applied throughout the brush and so the brush has a denser appearance.

**Opacity**
Opacity works with all brushes. By varying Opacity it is as if we have painted the brush on a layer above the image and then varied the transparency of the layer and then merged the two layers to achieve the final pixel color for each pixel.

The illustration above shows application of a brush where the Opacity of the top row is greatest (100) while the Opacity of the lowest row is only about 30. In all three rows the three yellow applications of the brush were made on top of the same cyan line. To allow comparison with Density, in each of the three rows the Density has been increased from left to right.

Note that Opacity is different than Density. The Opacity parameter applies evenly throughout the entire brush. By varying Opacity the entire brush effect is more or less transparent in an even way throughout the entire brush. Density changes the relationship within the brush.
Paint Bucket Tool

The Paint Bucket tool fills regions of similarly colored pixels with foreground color. There are four subtleties involved in using the Manifold paint bucket tool as compared to use of such tools in simpler graphics software:

- By default, the paint bucket tool pours color only into pixels contiguous with the point where the tool is clicked. Override this by using a **SHIFT** click.
- The paint bucket affects only selected pixels. If there is no selection, the paint bucket will operate on the entire image.
- The paint bucket pours color only into those pixels whose color values are within the **Tolerance** value set in the Tool Properties pane. Override the **Tolerance** consideration by using an **ALT** click.
- Use a **CTRL** click to paint color with the paint bucket while preserving the intensity levels of the previous colors. This is like a paint bucket "colorize" effect.

To use this tool we click on the Paint Bucket tool in the Tools toolbar for images. In the examples below we use a foreground color of purple. These examples will use the sample **bronze** image that has had the Threshold command applied to convert it to only black and white colors. This will show the effect of the paint bucket tool in a simplified image.

When the paint bucket is enabled Manifold changes the mouse cursor to a simple cross icon as shown above. We will click onto the black region at the rear of the monument.

The paint bucket pours purple foreground color into all black pixels that are contiguous with the pixel where the tool was clicked. Note that regions of black pixels in the image not contiguous with the click have not been colored.

**Using the SHIFT Key**
If we click at the same spot using a `SHIFT` click the paint bucket will pour purple color into all of the black pixels in the image no matter where they are located. This is a useful technique to replace one color with another throughout all pixels in the image.

**When a Selection is Present**

Suppose we select a rectangular region of pixels and `SHIFT` click with the paint bucket at the spot shown. The paint bucket tool will compute the pixels to be affected based on the color at the spot clicked but it will paint color only into those pixels that are within the selection.

If we switched foreground color to green and `SHIFT` clicked at the spot shown in the preceding illustration the paint bucket will paint green color into all of the black pixels within the selection.
**Tolerance Parameter**

The paint bucket tool measures the color value of the pixel at the point where it is clicked and then operates on pixels whose colors fall within the **Tolerance** parameter set in the Tool Properties pane. This is different than the default behavior of paint bucket tools within simpler graphics editors such as Microsoft Paint, where a paint bucket pours into all pixels of like color, or where the paint bucket pours into any selected region without regard to the color of pixels it is painting.

When painting into regions of pixels that are all the same color the use of the **Tolerance** parameter does not matter since in that case all of the pixels probably will fall within tolerance value being used. The **Tolerance** parameter most likely will make a difference when painting into regions of gradients or subtle transitions between colors.

Let’s look at an example using an image that contains a smooth gradient of color from blue to purple. Suppose we change foreground color to yellow and click the paint bucket tool at the spot indicated above. If we are familiar with the default behavior of paint bucket tools in simple graphics editors we might expect the yellow color to fill the entire selected region.

In Manifold, given typical default settings of **Tolerance** we might get the above effect. What happened?

If the **Tolerance** parameter in the Tool Properties pane is set to a low value, such as **10**, the paint bucket tool will paint only those pixels that are close in color to the one originally clicked. Blue colors close to that clicked were colored and blue colors transitioning to purple were also colored if they were not so purple that they fell outside the **Tolerance** range of **10**. Because the gradient in the example image was created using the Gradient tool, the transition in colors in the image occurs in a smooth diagonal gradation from upper left to lower right. The pixels affected by the paint bucket tool thus fall within an even, rhomboid region within the selection. The clearly defined edge of the region painted yellow by the paint bucket occurs because that is the edge at which pixels pass beyond the tolerance difference of **10**.
To color the entire selected region with the paint bucket we can simply increase the **Tolerance** parameter so that a wider range of pixels will be affected.

![Image showing the effect of increased Tolerance](image)

The result of clicking the paint bucket at the same spot with a **Tolerance** of 100 is that the yellow color is poured into the entire selected region. We can also get the same effect with an ALT click.

**The ALT Key**

Wishing to pour a paint bucket of color throughout a selected region is such a common operation that Manifold provides a faster way to do so than resetting the **Tolerance** parameter. Pressing the ALT key while clicking to pour the paint bucket will have the effect of pouring with maximum tolerance for that pour. The paint bucket will fill all selected regions regardless of their previous pixel colors. If there is no selection the entire image will be painted.

![Image showing the effect of ALT click](image)

Suppose we select a rectangular region of pixels in the sample bronze image and click with the paint bucket at the spot indicated, using a foreground color of yellow.
If the Tolerance parameter is at the usual setting of 16 or so, the paint bucket will fill pixels that are blue in color and close to the blue of the pixel that was clicked. Note that the paint bucket action is computed over the entire image but applied only within the selected region. Thus, the pixels on the other side of the box from the spot clicked are contiguous (considering the entire image) within the given tolerance to the spot clicked. Most of the blue sky would be painted were it not for the selection; however, the yellow color is painted only into the pixels that are within the selection.

Had we ALT clicked with the paint bucket at the spot indicated all pixels within the selected region would have been painted.

The CTRL Key

The CTRL key may be used when clicking with the paint bucket to preserve the intensity of colors while painting with the foreground color. The CTRL key may be used in combination with either the ALT key or the SHIFT key.

Suppose we select all pixels within the bronze monument and click at the indicated position with the paint bucket using yellow color.
A CTRL-ALT click will pour yellow color into the entire selection while preserving intensity levels from the colors previously in use. The effect is like doing a Colorize with the paint bucket.

Clicking at the same spot with just an ALT click will pour color into the entire selection without preserving intensity levels.

**Local Clicks Apply throughout the Image**

The paint bucket tool computes its action based on the spot that it is clicked regardless of whether this spot is inside or outside the selection. If a selection exists, the action of the paint bucket will be limited to the selected regions but it will also apply to all selected regions even if they are not connected with each other. This allows us to apply paint bucket action using the color of pixels at whatever location is easiest to click even if we intend the tool to apply only within the selection.

When used with the ALT key to paint throughout the selection, this means we can click very rapidly without having to worry about clicking within the selection.
If we **CTRL- ALT** click as shown above the paint bucket will pour color into all selected regions, preserving the intensity. Note that it does not matter whether or not we clicked inside or outside of any of the selected regions.

We would have had the same effect if we had **CTRL** clicked at that spot (or at any other spot) with very high **Tolerance** setting in the Tool Properties pane. Using the **ALT** key is a handy way of painting throughout all selected regions without having to be precise in our application of the paint bucket tool.

This effect of the Manifold paint bucket tool is different than what is encountered in some other graphics software, where a paint bucket tool applies to a region contiguous to where it is clicked. For example, clicking a paint bucket tool on the blue sky as shown above in some graphics editors would have resulted in a flood of yellow paint throughout all regions of the image **except** the selected areas. Clicking within any one of the selected regions would have painted only that region and not the other selected portions.

Manifold works differently because Manifold has a very strong set of selection tools that ordinary graphics editors do not have. Given Manifold's powerful selection methods the design philosophy of the system is that one makes a selection that is to be affected and then one applies tools to that selection. If we wanted to paint all the regions except those selected we would have taken a moment to use **Edit - Select Inverse** to invert the selection and then painted with the paint bucket. A strong array of tools for making selections, saving them and modifying them lets us choose exactly the region in which we wish our painting tools to apply.

**Notes**

Note that the use of **Tolerance** to guide the action of the paint bucket tool is not a negative thing: it is a powerful capability that allows the paint bucket tool to be used with fine control for replacing colors over a desired range of values. If we want the paint bucket to work without regard to **Tolerance** setting we simply **ALT** click.

When the paint bucket tool is in action, Manifold changes the mouse cursor to a simple cross icon. This is different than the cursor style used by some other software products for this type of tool. Other software programs will often use a small paint bucket icon for their mouse cursors. This has the advantage of showing what painting tool is in use, but such cursor shapes make it much more difficult to click with precision upon a desired spot. Manifold's cross cursor for the paint bucket tool makes it immediately clear where the tool is clicked.
Sometimes we would like to know in advance where the paint bucket will pour without having to Undo the operation. To see where the paint bucket will pour, use Select Touch to click on the spot where the pour will be made. Because Select Touch also selects pixels using the Tolerance set in the Tool Properties pane, it will select the same pixels that would be affected by a paint bucket pour. To avoid altering a pre-existing selection, save that selection first in the Selections pane so that it can be recalled after the Select Touch selection.

When using the paint bucket the CTRL key may be combined with either the ALT key or the SHIFT key. We can do a CTRL-ALT click or a CTRL-SHIFT click; however, the ALT and SHIFT key don’t make sense to use together since their combined action is equivalent to an ALT click.

**See Also**

See the example Painting into the Alpha Channel for an example of how the paint bucket tool may be used to paint into the alpha channel of an RGBA image. The paint bucket tool when used as the example shows is a very handy way of adjusting alpha transparency in partially transparent images.
Gradient Tool

The gradient tool paints all pixels with a color gradient from foreground color to background color. It will paint all pixels in the selection, or in the entire image if there is no selection.

Suppose we have an image that is all black color.

We will set foreground color to blue and background color to a gold-yellow color.

Click on the gradient tool in the Tools toolbar for images.

To paint a gradient into the image, we click where we want the gradient to begin and then drag and release where we want the gradient to end. In the example above we clicked in the upper left of the image and have dragged down and will release in the lower right.
There is no selection made in this image so the gradient tool will paint the entire image. All regions before the beginning click will be pure blue in color. All regions after the release click will be pure gold yellow. The regions in between will be painted in a continuous gradient of colors in a smooth mix from blue to gold. The regions “before” and “after” the initial click and final release are defined by lines perpendicular to the line of click and drag.

The result is that all pixels in the image are painted in either blue, gold or a mix of shades from blue to gold.

The gradient tool paints a gradient based on the initial click and the final release point. These do not have to be within the image. In the example above we clicked well outside the image and then dragged right and released outside the image as well.
The transition from blue to gold will be computed over the entire range of the click and drag but only that part of
the gradient that falls within the image will be painted into the image. Note that the image does not have any
pure blue or gold in it since the end points of the gradient click and drag were outside of the image.

We can force the gradient to transition from foreground to background color more rapidly by clicking and dragging
over a smaller distance. In the example above we have clicked and dragged upwards and slightly to the right
over a small distance.

The result is a rapid transition, so rapid that it appears mostly green with most of the image painted in either blue
or gold color.

When a Selection is Present
When a selection is present the gradient tool will work only within the selection. In the image above we have selected several round regions of pixels and one rectangular region.

We will switch to the **border** style of selection so we can see what happens inside the selection without the dense red dot selection pattern getting in the way.

We now click and drag with the gradient tool in the path indicated. Note that the points of click and release could be anywhere: there is no significance in the illustration above that the point of release happens to be within the selection in this particular example.
The result is that the gradient is computed as usual but it is painted only into those pixels that were selected.

We can disable use of selection color temporarily to see the result without selection borders. Using gradients within selections is a technique with an infinity of cool artistic possibilities.

We can take our sample bronze image and select all pixels outside the bronze monument. We use the dense dots selection style to make it clear which pixels are selected.
Once more we will switch to the border style of selection so that the red selection dots do not interfere with our view.

Let's change the foreground color to black and the background color to orange.

We now use the gradient tool from top to bottom as shown.

Viewing the result with selection color turned off, we can see that the gradient painted color from black to orange only into selected pixels. The effect is as if we see the bronze monument in front of a color gradient.

Notes

The examples above show the results of using the gradient tool when a selection is present. However, we would more frequently use such gradient effects within different images stacked in layers rather than within a selection in
only one image. This would allow us, for example, to independently manipulate the "cut out" of the bronze statue in one image/layer while we adjusted the gradient in a different image/layer.

This would come in handy if we would like to clean up the "halo" of lighter pixels in the bottom part of the bronze statue. These appear as a result of sloppy selection when we selected all of the pixels outside the monument. If the bronze monument were in a different image than the "background" gradient it would be much easier to clean up the lighter pixels.

See the example Painting into the Alpha Channel for an example of how gradients are used to paint into selections as well as how they can be used to paint into just one channel of an RGBA image. Gradients are a powerful way to specify alpha transparency so partially transparent images can appear to be "feathered" into a larger image composition.

### Transform Toolbar - Images

#### Target box  Operation box  Source / argument box

The Transform toolbar makes changes throughout the entire image. If a selection is present it will operate only on those pixels in the selection. If no selection is present it will apply to all pixels in the image.

The Transform toolbar consists of three boxes, from left to right: a **target box**, the **operation box**, and a **source / argument box**.

- **Target box**
  - Also known as the **scope box**. The pixels that will be affected, altered or which will control the operation. Choices in the target box will be [All Pixels], [Selection] or the names of any saved selections that have been saved in the Selections pane.

- **Operation box**
  - The function to be applied. The operation box is context sensitive and will show only those operations that make sense for images.

- **Source / argument box**
  - The value to be used. Depending on the operator, this may be another pixel set or a value entered by the user. Many operators, such as Desaturate do not require a source or argument. The source / argument box will not be enabled for such operators.

We use **dark blue**, **black**, and **violet** bold face fonts in this topic to distinguish the three boxes to make examples and explanation more clear. In real life Manifold uses the same black font color in all three Transform toolbar boxes.

#### Using the Transform toolbar

1. Click on the map layer or image that contains the target pixels.
2. Make a selection if the operation is to be applied just to the selection and choose [Selection] in the target box. Alternately, choose the name of a previously saved selection.
3. Choose the desired operator in the operation box.
4. Choose or specify a value in the source / argument box, if this operation requires it.
5. Press **Apply**.

#### Target box
The left-most box specifies the target pixels. These are the pixels that will be affected, altered or which will control the operation. The example above shows that there is no selection present, so the Transform toolbar will apply the Transform function to all of the pixels in the image. If we make a selection, the target box would include a [Selection] choice so that whatever operation we apply with the Transform toolbar will act only on those pixels that are part of the current selection. We could also choose the name of a saved selection.

Operation box

Choose a function from the long list of operators available in the operation box. The example shows Add Noise, which will add random noise pixels to the image. Note that many operators in the Transform toolbar's operation box are simplified versions of more flexible operators in the Image menu. The Transform Operators - Images topic lists operators available for use with images.

Source / argument box

Many operators do not require any sources or arguments. For these functions the source / argument box will be disabled.

Other functions will require specification of a source. This may an object set such as [All Pixels], [Selection] or the names of saved selections, all of which can be chosen from the combo box.

Some functions require an argument value that is entered by the user. To enter a value, click into the source / argument box, enter the value using the keyboard and then press Enter.

Examples

[All Pixels] Add Noise - Add random noise throughout the image.

[Selection] Add Noise - Add random noise only to those pixels in the selection.

[Selection] Brightness 64 - Reduce brightness to half the existing level (existing level is taken to be 128) for all pixels in the selection.

Sky Brightness 20 - Increase brightness for all pixels in the saved selection called Sky.

See Also

Transform Operators - Images
Many transform toolbar commands are versions of commands that also appear in the main Image menu. The hyperlinks below will jump to the Help topic for the corresponding command. Most commands are enabled for RGB or RGBA images. To use these commands with other types of images, first convert them to RGB or RGBA images.

- **Add Margin**
  Add to the rectangular size of the image by the given number of pixels at the margin.

- **Add Noise**
  Add color noise as set by the argument. A transform toolbar version of the Noise menu command.

- **Add Noise (Mono)**
  Add monochrome noise as set by the argument. A transform toolbar version of the Noise menu command with Monochromatic mode checked.

- **Auto Contrast**
  Automatically adjust contrast to balance contrast non-linearly throughout the entire image considering the overall intensity of each pixel.

- **Auto Level**
  Balance contrast throughout the entire image by balancing contrast within each channel individually.

- **Blur**
  Blur image by fixed amount. Blurs both features and colors. For a "blur" effect that does not mix colors, use the Median filters.

- **Blur (parameter)**
  Blur image by the number of pixels given in the argument. This blur is implemented as a matrix operator where the value of the argument is the weight of the center pixel with weights of all other pixels set to 1. Increasing the value therefore produces a lesser blur.

- **Brightness**
  Adjust brightness by the argument amount.

- **Contrast**
  Adjust contrast by the argument amount.

- **Crop**
  Crop image size to the selection.

- **Crop Margin**
  Crop the rectangular size of the image by the given number of pixels at the margin.

- **Desaturate**
  Reduce saturation to zero for all color values. Remains RGB.

- **Difference East**
  Detect and emphasize edge transitions to the East. See the Filters topic for general discussion of convolution matrix filters like this one.

- **Difference North**
  Detect and emphasize edge transitions to the North.

- **Difference North-East**
  Detect and emphasize edge transitions to the North-East.

- **Difference North-West**
  Detect and emphasize edge transitions to the North-West.

- **Difference South**
  Detect and emphasize edge transitions to the South.

- **Difference South-East**
  Detect and emphasize edge transitions to the South-East.

- **Difference South-West**
  Detect and emphasize edge transitions to the South-West.

- **Difference West**
  Detect and emphasize edge transitions to the West.

- **Diffuse**
  Migrate pixels by swapping them in a "random walk." Reapplying Diffuse several times will move pixels further and further away from their original positions.

- **Equalize**
  Equalize image to value given in the argument.
Flip Horizontally  Flip image left / right to mirror image. If a rectangular selection is present, the selection only can be flipped.

Flip Vertically  Flip image top / bottom to upside down image. If a rectangular selection is present, the selection only can be flipped.

Gamma  Alter gamma value by the argument.

Gaussian Blur  True Gaussian probability function blur by given amount. Slow to compute for very large images.

Grayscale  Cast all colors into gray tones by intensity. Remains RGB.

High Pass 1  Emphasize rapid transitions between pixels. Good edge detection with horizontal and vertical lines.

High Pass 2  Stronger emphasis on rapid transitions between pixels with strong edge detection on lines of all angles.

High Pass 3  Edge detection with crisp vertical and horizontals and fuzzy lines at other angles.

Invert  Invert pixel values so that 0 becomes 255 and 255 becomes 0. See the Invert topic for information on the Invert At transform.

Laplace 1  Good edge detection with emphasis on horizontal and vertical changes.

Laplace 2  Strongest changes at point of intersection of horizontal and vertical lines, resulting at a small X pixel pattern at the point of intersection; fuzzy detection of linear features at other angles.

Low Pass 1  Strong blur, with strong suppression of large pixel to pixel changes.

Low Pass 2  Blur, with medium suppression of large pixel to pixel changes.

Low Pass 3  Blur, with least suppression of large pixel to pixel changes. Removes graininess in some images.

Median Cross  Evens out color tending to preserve horizontal and vertical features. A "blur" that ignores perfectly horizontal and vertical linear features.

Median Square  Evens out color looking at a 3 x 3 matrix surrounding each pixel. A "blur" that preserves prevailing color.

Median Square 5  Strong evening out of color looking at a 5 x 5 matrix surrounding each pixel. Results in removal of isolated pixels of color differing greatly from surrounding pixels. A stronger "blur" effect that preserves prevailing color.

Motion Blur Diagonal 1  Blurs the image as if moved diagonally on the upper left / lower right axis by the specified number of pixels. See the Image - Motion Blur topic.

Motion Blur Diagonal 2  Blurs the image as if moved diagonally on the upper right / lower left axis by the specified number of pixels. See the Image - Motion Blur topic.

Motion Blur Horizontal  Blurs the image as if moved horizontally by the specified number of pixels. See the Image - Motion Blur topic.

Motion Blur Vertical  Blurs the image as if moved vertically by the specified number of pixels. See the Image - Motion Blur topic.

Posterize  Converts the image to a limited number of colors for a "poster" effect. Very low (2, 3 or 4) parameter values
Images gives the most obvious effect.

**Rotate** Rotate image given number of degrees. Negative degrees are taken as counter-clockwise rotation.

**Sharpen** Emphasize transitions to give images a sharper appearance.

**Sharpen (parameter)** A Sharpen with selectable value for degree of sharpness. This sharpen is implemented as a matrix operator where the value of the argument is the weight of the center pixel with weights of all other pixels set to 1. Increasing the given value therefore produces less sharpening.

**Sharpen More** Enhanced sharpness compared to standard Sharpen.

**Threshold** Converts an image into black and white based on whether each pixel is above or below the given value in intensity.

**Threshold Black** Like Threshold, but forces all pixels below the intensity given in the source / argument box into black while leaving all pixels above that value unchanged.

**Threshold White** Like Threshold, but forces all pixels above the given intensity into white while leaving all pixels below that value unchanged.

**Tile** Divides image into tiles of given X and Y extent and then averages pixel values within the tiles. This is used for aggregating and interpolating raster data images into larger tiles and for creating "pixelated" artistic effects.

**Tile Median** Equivalent to Tile command with Preserve Colors checkbox checked.

Many commands in the Transform toolbar are based on convolution matrix filters. See the Image - Filter topic for a detailed discussion of how matrices are used in filter effects and how to create new filter-based commands that may be added to the Transform toolbar.

**Image Menu Commands**

Many transform commands are one-line versions of commands available in the Image menu. The image menu appears when the focus is on an image window or an image layer is active in a map. Commands listed below will appear either under the main Image menu or under submenus such as Adjust or Effects. Commands are enabled for RGB or RGBA images. To use these commands with other types of images, first convert them to RGB or RGBA images.

**Open Palette** Enabled for palette images. Open the palette for this

**Dither** Uses a smaller number of colors in an image while preserving a perceptual effect through the use of dither patterns.

**Quantize** Convert image to the specified number of colors using enhanced or standard methods.

**Resize** Change the size of the image using desired interpolation method.

**Convert To** Convert images between grayscale, palette, RGB or RGBA images.

**Adjust Menu**

**Color Balance** Adjust RGB color values for Shadows, Midtones or Highlights

**Brightness / Contrast** Change brightness and contrast.
**Hue / Saturation**  
Alters hue, saturation and lightness.

**Invert**  
Invert pixel channel value. Creates a "photographic negative".

**Equalize**  
Alters individual pixel brightness so there is an equal number of pixels at all brightness levels.

**Gamma**  
Change mid-tone color brightness.

**Threshold**  
Forces grayscale pixels to white, black or unchanged based on a histogram.

**Threshold Color**  
Like **Threshold** but applies to each RGB channel individually.

**Posterize**  
Convert image to a limited number of colors for a "poster" effect.

---

**Effects Menu**

**Colorize**  
Adjust saturation and hue while keeping intensity the same, as if a color were applied to a grayscale version of the image.

**Gaussian Blur**  
Apply Gaussian bell curve blurring to create shadows, halos and transitions.

**Motion Blur**  
Streaked effect in the desired direction.

**Diffuse**  
Moves pixels by swapping them in a "random walk."

**Filter**  
Applies convolution filters, either custom filters or a choice from numerous presets.

**Fluoresce**  
Alter saturation and lightness in asymmetric ways to cause some colors to appear unusually bright.

**Noise**  
Add random monochromatic or color pixels.

**Relief**  
Add highlights and shadows to give a 3D effect to 2D images.

**Simplify**  
Aggregates pixels into larger clumps for a pointillist or mosaic tile effect. Can also remove small dots in cartographic work.

**Tile**  
Convert image into rectangular tiled regions of appropriate color.
**Transform - Add Margin / Crop Margin**

The **Add Margin** image transform toolbar operator adds the specified number of pixels to the top and sides of an image. By default, these are added as black pixels by default. The **Crop Margin** command

**Crop Margin** was used to trim 20 pixels from the margin of the center image to create the image on the left. To create the image on the right, we've added 10 pixels to the margin of the central image. By default, the **Add Margin** operator adds colors in whatever color is currently set as the foreground color in the format toolbar for images. Un-checking the **Fill new pixels with current color** box in Tools - Options will cause the margin to be added using invisible pixels regardless of the current foreground color.

Note: When used with surfaces, the **Add Margin** transform always adds invisible pixels.

**Use with Selections**

Selections do not affect the **Add Margin** or **Crop Margin** commands since these operate on the overall image rectangle.
Transform - Auto Contrast

The Auto Contrast transform operator automatically adjust the contrast throughout a given image or selection. If a selection is present, it works only on the selected pixels. If no selection is present it works on the entire image.

The Auto Contrast operator changes contrast throughout the image on a non-linear basis to achieve what the algorithm thinks is an optimum, even contrast. It works by reckoning the intensity of pixels considering all channels taken together. The results are similar to those produced by the Adobe PhotoShop Auto Contrast command.

The image on the right above has had Auto Contrast applied.

Note

Because Auto Contrast uses a non-linear algorithm that clips extreme bright and dark sample pixels, the effects it produces cannot be obtained by simply moving the Contrast slider in the Brightness / Contrast command.

See Also

See the Auto Level topic for a comparison of Auto Contrast and Auto Level effects. Auto Level adjust contrast within each channel individually.
Transform - Auto Level

The **Auto Level** transform operator is similar to Auto Contrast, except that whereas **Auto Contrast** sets contrast by reckoning pixel intensity considering all channels taken together, **Auto Level** sets levels within each channel separately.

**Auto Level** will often produce better contrast than either **Contrast** or **Auto Contrast**. However, because it adjusts intensities in each channel separately to produce the overall effect it can make minor color shifts. These minor color shifts will often enhance the sense of reality and vividness of an image.

**Example**

If take our standard **schloss** image above and apply **Auto Contrast** we get the image below.

If we start with the **schloss** image and apply **Auto Level** we get the image below. Compare the difference in appearance of trees in the image above with those in the image below. The **Auto Level** image provides slightly greener trees. There is also a greater sense of color differentiation in the yellow crenellations of the uppermost towers.
Compared to the original image the **Auto Level** image has slightly different color tones but in a productive way that yields a more vivid image. In other images, the color shift may be objectionable. For such images **Auto Contrast** may be a better choice.
Transform - Crop

The **Crop** image transform toolbar operator crops the image down to the minimum enclosing rectangle of the target box in the Transform toolbar. Normally, this is the **[Selection]** or the name of some saved selection. **Crop** has no effect if no selection is present.

The example above selects Tokyo airport and then crops down to the size of the selection.

**Crop** works fine even if the selected pixels are in many different discontinuous regions.
Classic mouse cursor "box" crops may be easily achieved using **select box** mouse selection. If desired, the selection style can be set to **border** to show the selection as a box.
**Transform - Desaturate**
Desaturates all colors in the image to produce an RGB image that uses shades of gray. Note that desaturation is not the same as conversion to a **Grayscale** image.

**Grayscale** and **Desaturate** will both produce monochrome grayscale images; however, the two have different effects as may be seen by applying them to the color wheel.

Applying **Grayscale** results in visual distinctions between the hues as color tones:

Applying **Desaturate** results in an even gray tone because the brightness level for all hues is the same:

When converting color images to photographs, **Grayscale** will usually produce a more natural appearance, while **Desaturate** will produce a more "posterized" look.

See also: Colors as Hue, Saturation and Brightness, Hue / Saturation
Transform - Diffuse

The **Diffuse** operator moves pixels by swapping them in a "random walk." Reapplying **Diffuse** several times will move pixels further and further away from their original positions. **Diffuse** is used to apply artistic effects.

The above images show a zoomed-in view with one application of **Diffuse** as well as a view after several applications of **Diffuse**. Note how pixels "drift" at random from their original positions with repeated applications of **Diffuse**.

When used with photographic images, **Diffuse** provides a pixelated, "dry brush" painterly effect. The illustration above shows the effect after using a level of 9, applied to the sample **bronze** image that was previously altered using the fluoresce command.
If a selection is present, the effect can be restricted to the selection as seen above.
Transform - Flip Horizontally / Flip Vertically

The Flip Horizontal and Flip Vertical commands flip an image right/left (mirror image) or vertically up/down. If a rectangular selection is present, it may be used in the scope box and the flip will be applied to the selection only.

For example, if we start with a selection in the example bronze image (shown using a border selection style) and apply Flip Horizontal to the selection it will flip the selected pixels horizontally.

Applying Flip Vertical will flip the pixels vertically.
Transform - Grayscale
Casts all colors into gray tones using computed intensities of colors to choose gray tones for pixels. Note that grayscale is not the same as conversion to a Grayscale image

Grayscale and Desaturate will both produce monochrome grayscale images; however, the two have different effects as may be seen by applying them to the color wheel.

Applying Grayscale results in visual distinctions between the hues as color tones:

Applying Desaturate results in an even gray tone because the brightness level for all hues is the same:

When converting color images to photographs, Grayscale will usually produce a more natural appearance, while Desaturate will produce a more "posterized" look.

See also: Colors as Hue, Saturation and Brightness, Hue / Saturation
Transform - High Pass
The High Pass filters in the Transform toolbar for images detect rapid changes in pixels using three different analytic matrices. These filters will emphasize linear features such as edges.

The examples below are based on an example image that is a screen shot of a drawing, converted to a raster image and shown above. The image therefore provides a nice mix of strong horizontal and vertical linear features made up of evenly aligned single pixels as well as pixels arranged in uneven linear patterns.

High Pass 1 - Good edge outlining with emphasis on horizontal and vertical changes.

High Pass 2 - Stronger emphasis on edge changes at all angles.

High Pass 3 - Strongest changes at point of intersection of horizontal and vertical lines, else relatively unchanged horizontal and verticals with fuzzing of linear features at other angles.

See the Image - Filter topic for a detailed discussion of how matrices are used in filter effects.
No Data
Transform - Median Cross / Median Square

The Median Cross, Median Square and Median Square5 filters in the Transform toolbar for images use a convolution matrix filter to even out colors and reduce smaller details while preserving prevailing color. These transforms are useful for processing images that will be used for classification and vectorization, to remove extraneous noise and detail while preserving broad regions of color.

- **Median Cross**: Evens out color tending to preserve horizontal and vertical features. A "blur" that ignores perfectly horizontal and vertical linear features.
- **Median Square**: Evens out color looking at a 3 x 3 matrix surrounding each pixel. A "blur" that preserves prevailing color.
- **Median Square5**: Strong evening out of color looking at a 5 x 5 matrix surrounding each pixel. Results in removal of isolated pixels of color differing greatly from surrounding pixels. A stronger "blur" effect that preserves prevailing color.

The effects may be seen by applying the transforms to an overhead image of Tokyo airport, seen below.

The **Median Cross** transform reduces speckles and averages out detail in large areas of color while preserving major lines such as the edges of the land areas against black water areas. Note that some fine details such as the very thin diagonal line extending into the water in the upper center of the image is preserved.

The **Median Square** transform seen below creates a greater averaging and greater suppression of speckles. The thin line extending into the water has been averaged away, but the small white dot indicating an island or large ship in the channel in the lower left corner is still present.
The **Median Square5** transform has the strongest averaging and nearly total suppression of speckles and individual dots.

Compare the above image to the image below created using Gaussian Blur with a setting of 5 pixels.

Note that the **Median Square5** transform preserves object outlines while **Gaussian Blur** simply blurs everything together within a range of 5 pixels.
**Transform - Rotate**

The **Rotate** transform rotates an image or surface the number of degrees specified in the source / argument box. Positive degree numbers cause a clockwise rotation and negative degree numbers cause a counter-clockwise rotation.

For example, rotating the above surface by **-45** degrees results in the surface below.

Rotation parameters may be decimal fractions, such as **45.55** for more precise rotations, if desired.
No Data
Maps

Map windows are the most frequently used windows in Manifold. A map is a projected view that contains layers. Each layer is a separate drawing, image, surface or labels component. Maps are the only view windows in Manifold with layers. They are used to see drawings, images, surfaces and labels components in combined, layered views.

Creating a New Map

We can create a new map using whatever source materials we like:

1. Create a new project using File - New. This opens a project view window.
2. Using File - Import, import any drawings or other components into the project the new map will use.
3. Create a new map in the project using File - Create - Map. We must specify the name of at least one drawing or image this map will use. In the Create Map dialog, check all the drawings or images from the project you would like to appear in this map.
4. Double-click on the new map in the project view window to open it in a map view.
5. Drag and drop any additional components from the project view window into the map view window. This adds them to the map.
6. If images (or any other components) are not already georeferenced, they must first be georegistered so that they appear in the proper geographic location at the correct size.
7. If desired, use Edit - Assign Projection to change the projection used by the map.

Maps are also used to edit drawings and images by using multiple layers for edits. PhotoShop-style editing of images using layers is accomplished in Manifold in a map view.

There can be more than one map window open and each map window is fully independent. Drawings and images can be used in multiple maps at the same time.

Maps can have multiple layers where each layer is a different drawing, image, surface or labels component. Tabs at the bottom of the map window allow moving layers back and forth in the layer stack. Double-clicking on a layer tab turns that layer on and off for visibility. Right-clicking on a layer tab pops open a menu of commands for that layer.

Editing an image or drawing in one map view will simultaneously change it in all other map or drawing or image windows in which it appears.

There is no structural limit on the number of maps or the components they show in Manifold. Since processing overhead increases as many maps are opened using many drawings and images, at some point a practical limit
will be reached based on your computer’s speed and your patience for slower operation. However, for ergonomic efficiency it makes sense to avoid a display cluttered with too many different maps showing many different views.

**Menus**

When we click into a map window, Manifold will automatically change menus and toolbars so they contain those menus and controls that work with maps. When we click on a layer in a map to make it active, the menus and toolbars will adjust to the configuration for that type of component.

Menu items that appear when a map window has the focus will apply only to the active layer. For example, **Edit - Select All**, **Select None** and **Select Inverse** apply to the active layer only, as does **Edit - Copy**. To apply selection commands to all visible layers, use the **ALT** key together with a mouse-based selection command such as **Select Box** or with a keyboard short cut such as **CTRL- C**. For example, and **ALT-Select Box** used with **Invert** selection mode will create an inverse selection over all visible layers. **ALT-CTRL-C** will copy all selected objects from all visible layers.

**Projections**

Map views use projections. At any point the map may be transformed to show its contents in a different projection.

When using a projection, map views do not change the native data within a drawing or image. Instead, the contents are reprojected “on the fly” if the projection shown in the map view is different than that specified within the drawing or image. If the same drawing is used in different maps, those different maps can use different projections and thus show that drawing simultaneously in different projections.

Since the mathematics of projection require a lot of computer processing, if using projections within maps that contain large drawings or images it makes sense to re-project the drawings and images used into the projection that will be used. If a drawing or image already uses the desired projection, there is no need for the map view window to reproject it on the fly. The result will be much faster interactive operation.

When a map is first inserted into a project, if it is created using only one component the projection of that drawing or image will be used for the map. If a map is created using multiple components at the same time, then Manifold will use the projection of the **largest** image or surface as the default projection. This is because images and surfaces are slower to re-project than drawings so the map will run faster if it takes the native projection of an image or surface and then re-projects any drawings to match.

At any time we can change the projection used by a map using the **Edit - Assign Projection** dialog.

**A shortcut:** If we would like the map to use a specific projection that's used by one of the components used as a layer in the map, we can easily specify the projection to be used by right-clicking on any layer and choosing the **Use Projection** choice from the context menu. The map will automatically switch to using the native projection used by that layer.

**Exporting Maps**

Manifold maps can contain many different types of components, such as drawings, images, labels and surfaces. Since most GIS formats can host only one such component type and do not support layers, the usual way of exporting a map to some different format is to export on its own each component that participates in the map. For example, various drawings and images in the map could each be individually exported on their own. An exception to this is the ability to export drawings in a map to either **DXF** or **KML** format.

Maps may be exported to **AutoCAD DXF** files, excluding any surface or image layers. See the Export Drawing - **DXF** topic.

Drawing layers in maps may also be exported to **Google Earth KML** or **KMZ** files. See the Export Drawing - **KML**, **KMZ** topic for details.
Layers
Maps show drawings, images, surfaces and labels components stacked in layers.

The "empty space" in drawings and labels components is transparent so objects in lower layers can be seen. Any layer may be made partially transparent so items underneath the "solid" parts of the layer may be seen as well.

Layer Tabs
Maps show layer tabs at the bottom of each map window. The layer tabs are automatically named the same as the drawing or image they contain. The leftmost tab is the top layer. Layer tabs may be used to manipulate the layers in a map.

Layer Tab Controls

- Click on a layer tab to make that layer the active layer.
- Double click on a layer tab to turn that layer on / off for display.
- Click and drag on a layer tab to move that layer to a different position in the layer stack.
- Right click on a layer tab to pop open a context menu of things we can do with layers.

The active layer tab is white. Inactive layer tabs will appear in the color defined in our Windows Desktop Display settings for “3D Objects.” Only one layer is active at a time. Selection commands and windows clipboard commands operate on the active layer only by default.

The Layers Pane

The Layers pane provides an alternate user interface for working with layers. The layer tabs at the bottom of each map are convenient when a small number of layers is used in a map. When a map has many layers the layer tabs are inconvenient to use because we must spend a lot of time scrolling left and right through the layer tabs to get to any particular tab.

The Layers pane provides a list of all layers in a map. It is normally left open and docked to the left or right side of the main Manifold window in a vertical orientation. It may therefore be used to keep dozens of layer names in view and to rapidly click on any one of them. For intensive image editing work involving many layers it provides faster control over layer characteristics (such as opacity) as well.

Layers Pane Controls

- Double-click on a layer to highlight it and to make it the active layer.
- Click on a layer and press the Switch to button in the layers pane toolbar to make it the active layer.
• Use the checkboxes to turn layers on and off for display.
• **Right click** on a layer for a context menu for that layer.
• Each layer may be set to a percent opacity to specify partial layer opacity.
• Move layers up and down in the layer stack with the **Move Up** and **Move Down** buttons.
• The **Border** and **Background** may be turned off and on using their checkboxes.

### The Border and Background Layers

The layers pane will always show two additional layers: the **Border** layer and the **Background** layer. These are system layers and are not real "layers" in that they do not correspond to any drawing or image in the project. Both may be switched ON or OFF via their checkboxes.

The **Border** layer shows a border about the maximum horizontal and vertical extent of the map. It is a useful way of checking if any objects are in the map but are not in view.

The **Background** shows what color to put underneath all other layers. By default white, it may be switched off to easily see if any white space in the map is part of the background or not. The background color for all components is set by default in the Tools - Options dialog. Each individual component can have its background color set to some color different than the default by opening the component in a window and choosing **View - Properties**. Changing the background color can dramatically change the appearance of a map.

The following three illustrations show the effects of changing only the background color from the default white, to black to green.
It is obviously important to consider background color when choosing formatting colors for objects in drawings. Note, for example, how the black color used for a railroad disappears when black is the background color. The map shown above was intended for use with a green background color, and not with white or black background. (The illustrations show a small region in Texas taken from a declassified US Military map provided by NIMA in VMAP Level 2 format.)

**Using Layers to Set the Map Projection**

When a map is first inserted into a project, if it is created using only one component the projection of that drawing or image will be used for the map. If a map is created using multiple drawings or images at the same time, then a random choice of one of the native projections of one of the constituent components will be used.

At any time we can change the projection used by a map using the Edit - Assign Projection dialog. If we would like the map to use a specific projection that's used by one of the components used as a layer in the map, we can easily specify the projection to be used by right clicking on any layer and choosing the **Use Projection** choice from the context menu. The map will automatically switch to using the native projection used by that layer.

**Layer opacity**

Layers may be given an opacity percentage that applies to the entire layer. As opacity ranges from zero percent to one hundred percent the layer will be either completely transparent or totally opaque.

If an image layer contains an RGBA image where individual pixels may have different degrees of transparency, the layer opacity will be evenly combined in a proportional way with all pixels.
Opacity Dialog

An alternative to setting layer opacity via the % edit box in the Layers pane is to use the opacity dialog that pops up from the context menu for map layer tabs. Right click onto a layer tab and choose Opacity to open the dialog:

The opacity dialog allows interactive setting of opacity by moving a slider bar. Check the Preview box to see the effect in real time. We can also click into the edit box and change the opacity value number.

Context Menu for Map Layer Tabs

Right click on a layer tab to see the context menu for that layer:

Visible    Show or hide this layer.
Add        Add a new blank drawing, image or labels component using the map's projection, or launch the Layers dialog to add components from the project pane. The component will be added as a layer just above the active layer tab.
Cut         Windows clipboard cut operation. Copy objects or pixels to the clipboard and delete them from the drawing.
Copy        Windows clipboard copy operation. Copy objects or pixels from this layer onto the clipboard.
Paste       Paste the contents of the Windows clipboard into this layer replacing any selected items.
Paste Append Paste the contents of the Windows clipboard into this layer.
Delete from Map Delete this layer from the map. The component will continue to exist in the project pane and in any other maps in which it participates.
Duplicate   Make a copy of this component in the project pane and insert it in the map.
Open        Open this component in its own window. If it is already open in its own window, activate that window.
Open in New Window Open this component in its own window. If it is already open in its own window, open it again in yet another window.
Rename      Change the name of this component
Order       Move layer to top, up, down or to bottom of layer stack.
Opacity     Set the opacity of this layer. See the Layer Opacity topic and the note below regarding the Opacity dialog.
Restrictions Specify if this layer allows clicking, editing or selecting using mouse commands. See the Layers pane topic for more information on layer restrictions.
Center      Center the active layer in the map window.
Maps

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zoom</td>
<td>Zoom to fit the active layer in the map window.</td>
</tr>
<tr>
<td>Zoom to Selection</td>
<td>Zoom to fit the selection in the active layer in the map window.</td>
</tr>
<tr>
<td>Match</td>
<td>Enabled for image or surface layers. Crop / expand / resample / re-project the active layer so that matches the image or surface specified. Often used to prepare an image so that it is the same size and coordinate system as another for use as a mask or other purposes.</td>
</tr>
<tr>
<td>Register</td>
<td>Move and rescale this drawing to specify its geographic location. A manual operation that is rarely used. See the Georegistration topic for easier methods.</td>
</tr>
<tr>
<td>Project to Map</td>
<td>Re-project this layer’s component to match the projection currently used in the map. This is a permanent alteration in the data.</td>
</tr>
<tr>
<td>Transfer Selection</td>
<td>Transfer selections between components.</td>
</tr>
<tr>
<td>Use Projection</td>
<td>Apply the native projection of this component as the projection used by the map.</td>
</tr>
<tr>
<td>Properties</td>
<td>View the properties dialog for this layer.</td>
</tr>
</tbody>
</table>

Example

Manifold System has such powerful image manipulation capabilities that it may also be used to create complex images by combining multiple images in layers as is done with professional quality graphics image editors such as Adobe PhotoShop.

For example, the marketing promotion "splash" screen used with 5.x and 6.x series Manifold System releases was created using over 40 layers of other images. Alpha pixel transparency and invisible pixels were used throughout the various layers so lower layers could combine to form the image. Looking at seventeen of the various layers we can see how the image was created:
The bottom three layers were screen shots from a Terrain window using different settings for sky and surfaces. The uppermost of these is the grid.

The next two layers are the floating city and its shadow, which are in separate layers so that the opacity and relative position of the shadow may be easily controlled. See the Gaussian Blur topic for details on creating drop shadows. The next twelve layers consist of text or logotype images together with various drop shadows.

The actual production "map" for the splash screen included over 40 layers. The extra layers contained variations of the logotype, different visual elements such as different sky cloud patterns and other variations. By checking the layers ON and OFF and moving the various layers about it was easy to try out many different variations to end up with exactly the effect desired.

Each image in the layer stack was heavily manipulated using various effects, resized, partially erased or made transparent to "feather" its features visually into features from other layers, and so on. For example, the city originated in a scanned photograph snapped from the bell tower in St. Mark's square in Venice. It was cut out of the photograph and then different parts of the image had hue, contrast and color balance adjusted in different ways. To bring out detail in the trees the green areas were re-built from several layers using different settings for brightness, contrast, saturation, colorization and color balance.

See the Composing Complex Images in Layers topic for an additional example of how to compose complex images using layers in Manifold.
Layers and Commands

Commands in the map window will automatically adapt to the type of layer that is active. The active layer is the layer whose tab was last clicked. Click on a layer tab to make it the active layer. The active layer’s tab will be shown in white. Toolbars will switch to supporting whatever type of component (image, drawing, labels or surface) is in that layer. Commands that are available in the component’s native window will be available for that component when it is the active layer in a map.

We can summarize the action of commands on layers in maps:

- The results of drawing transform operators always will be created in the active layer.
- Drawing transform operators use whatever layers or selections are specified in the scope box.
- Menu-based selection commands always work on the active layer.
- Mouse-based, visual selection tools work only on the active layer by default but can optionally apply to all layers.
- Windows Clipboard commands (including the Delete button) by default work only on objects in the active layer but optionally can apply to all layers.

ALT Key in Maps

Selection and Clipboard (Cut, Copy, etc) operations in maps work on the active layer by default. To extend the action of these operations to all visible layers, press the ALT key. For example, drawing a selection box in a map will only select objects in the active layer. Holding the ALT key while drawing a selection box will select objects from all visible layers. Pressing Delete deletes selected objects in the active layer only. Pressing ALT-Delete deletes selected objects in all visible layers. Think of ALT as a mnemonic for “All.”

Menu items that appear when a map window has the focus will apply only to the active layer. For example, Edit - Select All, Select None and Select Inverse apply to the active layer only, as does Edit - Copy. Menu commands cannot be extended to include all layers. The ALT key works only with mouse selection methods and keyboard commands. It does not modify menu commands (such as Edit - Copy) which always work on the active layer.

To apply selection commands to all visible layers, use the ALT key together with a mouse-based selection command such as Select Box or with a keyboard short cut such as CTRL-C. For example, an ALT-Select Box used with Invert selection mode will create an inverse selection over all visible layers. ALT-CTRL-C will copy all selected objects from all visible layers.

Example

We will import the bay_hydro, bay_land_areas and bay_roads drawings from the Examples directory on the Manifold CD.
We will rename the drawings to roads, hydro and land for brevity and then create a map with them as seen above. We will also open each drawing in its own drawing window. Note that the land layer is the active layer in the map. Our objective is to select all the points and to delete them.

Configure the selection toolbar as shown above. We choose Replace as the selection mode, and enable Select Points, disable selection of areas and lines, and choose Select Box. Note that the Select Labels and Select Pixels buttons are not enabled because there are no labels, image or surface components in the map.
While holding down the \textit{ALT} key, we click and drag in the map to open a selection box that encompasses all of the objects in the map. Because the \textit{Select Points} filter button is in, the \textit{Select Box} command will select all points within the box. The \textit{ALT} key extends the action of the command to all visible layers. If we had not used the \textit{ALT} key the command would have applied only within the active layer.
Points will be highlighted in red selection color. Note that the focus is on the map window as can be seen by the active color in the map window's title bar.

Even though there are selected objects visible in the map, if we choose Edit in the main menu with the focus on the map window, none of the Windows clipboard (Cut, Copy, etc.) commands are enabled. The reason is that there are no selected objects in the active layer.

We can click on the hydro layer to make it the active layer.
The **hydro** layer's tab changes to white to show it is the active layer.

If we now choose **Edit** in the main menu, we will see that the Windows clipboard commands are enabled. This is because there are selected objects (points) in the active layer.

If we choose **Edit** - **Delete** we delete the selected points in the active layer. Note that the **hydro** points disappear both from the **hydro** drawing window as well as from the map window.
To delete the points from the roads layer we click on the roads layer tab to make it the active layer and then we choose Edit - Delete.

See Also

Layer Restrictions
Layer Opacity
Zoom Ranges
Layer Restrictions

The Restrictions button in the layers pane displays and sets layer restrictions. Layer restrictions may also be set by right clicking on a layer tab in a map window and choosing Restrictions.

Each layer within a map may be specified to be clickable, editable, selectable or snapable (in any combination) with mouse commands. Each layer has four restriction options:

- Enable mouse clicks
- Enable mouse editing
- Enable mouse selection
- Enable mouse snaps

By default, all four options are enabled so objects in layers by default may be clicked, edited, selected or snapped.

The layers pane shows edit restrictions with a no marking each restriction. The "yes" implied by default is not shown for greater legibility. For example, in the above illustration the Cities layer is clickable and snappable but not editable or selectable. Double-click into any restriction setting in the layers pane to toggle it between no and the default yes setting.

If we were to right click on the Cities layer tab in the map and choose Restrictions the Layer Restrictions dialog would show restrictions status for that layer. In this dialog a check mark enables the function and no check mark removes the function. For example un-checking the Enable mouse edits box will prevent a CTRL-ALT click on an object in that layer from selecting it for editing.
Layer restrictions are used to show layers in maps while preventing some layers from participating in mouse commands. This makes it easier to edit desired layers in complex maps. Another use for layer restrictions is in Manifold IMS to restrict the action of hyperlinks or the Info tool to a limited set of layers.

Layer restrictions operate only with mouse commands. Keyboard and menu commands are not limited by layer restrictions, and smart mouse selection does not observe layer restrictions. For example, if a layer has had the Enable mouse selection checkbox un-checked none of the mouse selection commands will select objects in that layer. However, using CTRL-A or Edit - Select All will select all objects in that layer, the query toolbar will select objects in that layer and queries will select objects in that layer.

**Copying Restrictions to All Layers**

A CTRL-double click (that is, double-clicking while holding down the CTRL key) onto a restriction for one layer in the layers pane will copy that restriction to all layers. For example, suppose we have many layers in a project but we only want one layer, called Roads, to be editable. We can set the Editable restriction in the Roads layer to no and then CTRL-double click to set the Editable restriction to no for all layers. We can then click off the Editable restriction in the Roads layer so that it is the one layer that is editable while all the others are not editable.

**Notes**

We should keep in mind the interplay between user interface dynamics, layer structure and layer restrictions. When computing the effects of clicks and layer restrictions, Manifold first determines the clicked object and then checks if the click should be processed or not.

Let's consider an example case of a map with three layers. The upper layer has a single point formatted as a green circle, the middle layer has an area and the lower layer has a point formatted as a green square.

![Map View Example](image)

Seen in the map view, we see the round point above the area, shown with a hatching area style with transparent color in background, so that anything below the area shows through the "mesh." The square point is below the area.
If we could see the layers in the map in three dimensions, they would be stacked as above. Suppose the points are in layers with no restrictions, but the area is in a layer that is restricted and does not allow clicks.

In that case we could click on the round point but we could not click on the square point. Because the (restricted) area is above the square point, if we tried to click on the square point we could not do so because any clicks in that vicinity would be judged as clicks on the area that covers the square point.

Suppose the square point was in a layer above the area. Even in this case it would be difficult to click on the square point because anytime we click near the square point but not right on the center of the square point Manifold will judge the click to be onto the area. Since the area is within a restricted layer the click would not be processed.

Note that the size of the icon used for a point style does not determine the clickable region around that point. One must click on the center of the point to select it.
See Also

Selection
Editing with Snap
Edit - Snap To
Layer Opacity

Each layer in a map can have the opacity of that layer individually set from zero opacity to 100 percent opaque. Layers that are partially transparent will allow objects, labels and pixels from layers underneath them to be partially visible. Opacity is controlled by settings in the layers pane. Transparency may also be set via the Opacity dialog in the context menu for layer tabs in maps.

To change the opacity of a layer in a map, open the map and then open the layers pane. Press in the View Opacity button in the layers pane. This adds an Opacity column to the layers pane.

Double-click into the Opacity setting for any layer to change it. 100% opacity (default) means the contents of the layer are opaque. 0% opacity means the contents of the layer will be invisible because they are completely transparent. When entering opacity numbers, there is no need to enter the % percent character: simply enter a number from 0 to 100 and press ENTER.

An alternative to setting layer opacity via the % edit box in the Layers pane is to use the opacity dialog that pops up from the context menu for map layer tabs. Right click onto a layer tab and choose Opacity to open the dialog:

The opacity dialog allows interactive setting of opacity by moving a slider bar. Check the Preview box to see the effect in real time. We can also click into the edit box and change the opacity value number.

Layer opacity will be combined with whatever opacity is already defined within the component. For example, RGBA images will have per-pixel transparency enabled. Using layer opacity with such an image layer will apply the layer opacity throughout the entire RGBA image in addition to any per-pixel transparency.

Note that changing the transparency of a map’s layer in the layers pane does not make any changes to the component involved. It only changes how that component is displayed in the layer stack. So, for example, if we have an RGBA image participating in a map when we change the layer opacity there is no change to the alpha channel values for each pixel.

Examples

Suppose we have a drawing of US counties (at left, below) and a drawing of congressional districts (at right):
Both drawings consist of areas. We would like to show them together in a map so that we see how congressional districts are shaped compared to counties. Without layer opacity it would be difficult to portray the overlap cleanly since whichever drawing was placed in an upper layer would hide the lower drawing.

Our solution is to use layer opacity. We will place the congressional districts drawing in an upper layer and then change its opacity until we get the desired effect showing both layers at the same time.

90% opacity: This provides slight visibility of the underlying counties drawing layer and would be a good choice if the main interest is the congressional district outlines.

75% opacity: Allows more of the counties drawing to show through the upper layer. Note that combining the colors of the lower and upper layers will tone down the rather garish initial colors of the upper layer. In effect, we are adding gray color to the upper areas which has the effect of reducing saturation of the combined image.
50% opacity: A good choice if the main focus is on county boundaries but where congressional district shapes should be easily available.

25% opacity: This decreases opacity of the upper layer so much that it is barely visible. This is a good way of applying subtle color to a map intended mainly to show counties that may also need to have other information, such as labels or point locations, included as well. The relatively faint outlines of congressional boundaries provide guidance for those interested in districts without being so visually obvious that they clutter the map.

Layer opacity is a great tool for utilizing images within maps.

A classic problem with georegistered images displayed in the same map as drawings is that the visual context of the image may interfere with the clear visual perception of objects in drawings.
Decreasing layer opacity makes the image fainter so that drawing objects can better stand out. A similar effect could be achieved by altering the brightness and contrast and saturation of the image but that would require an alteration in the image.

Changing layer opacity allows us to increase or decrease opacity of the layer without any change in the image component. This allows us to interactively seek the exact visual effect desired by increasing and decreasing the opacity for the layer. We will often use layer opacity in this way to "fade out" a background map when we would like to provide a subtle backdrop while retaining visual attention to map elements in the foreground.

Making images partially transparent allows objects in drawings underneath those images to be visible. In the above screen shot, we see two green circular areas appear through the image from a layer below.

The layers pane for the above map sets the opacity of the SanFrancImage layer at 50% so that the circular areas in the Antennas drawing can be seen.

Note that layer opacity is applied in addition to any transparency effects inherent in the image itself.
For example, suppose we change the **SanFranImage** image into an RGBA image and then "feather" the transparency so that all but the central portion of the image is transparent. We can use the image in a map with drawings. This example shows the same drawing layers with a new image layer. (The **Antennas** drawing is clicked off in this illustration).

If we apply transparency to the image layer and turn on display of the **Antennas** drawing we can get an effect that combines the pixel transparency in the RGBA image with the overall layer opacity given in the map's layers pane.

Layer opacity works with any layer that may be displayed in a map. Labels layers, for example, may also be made partially transparent in maps as shown above.

**Graphics Arts**

Combination of layer opacity with RGBA pixel transparency provides a wide range of artistic effects.
We can copy our standard bronze example image, convert it to an RGBA image and then create a feathered circular region while leaving the rest of the image transparent.

We can then layer this in a map with a copy of the bronze image that has opacity set low to provide a fainter image.

The result is a combined effect that highlights only part of the bronze image. The effect is more dramatic when seen in a True Color 24-million color image as is normally seen in monitors. The images in this Help file have been artificially reduced to only 256 colors to fit within the constraints of the Microsoft Help system.

Note that this same effect could have been achieved by feathering the image in the upper layer alone. In this case, instead of having just an opaque circular region feathering to the larger part of the image with complete transparency, the circular region could have been feathered out to a larger region of partial opacity. The advantage of creating the combined image in two layers is that the bottom layer can easily be altered in opacity to allow an interactive approach to setting the opacity desired. One simply increases or decreases the opacity of the bottom layer until one sees the effect desired.
Combined Layer Opacity

Layer opacity in Manifold works just as transparency does when stacking layers of transparent plastic film. The results are visually combined and not simply arithmetically additive.

Suppose we have a map with three drawing layers, each of which contains a blue square. The screen shot above shows the map. Two of the layers have layer opacity set to 50% and in these two layers the blue squares have been positioned to overlap each other. The third layer has 100% percent opacity.

Note that the square in the layer with 100% opacity appears completely opaque. The checkerboard background is not visible through it at all. The squares in the partially opaque layers have been rendered partially transparent so that the checkerboard background may be seen through them. Where the squares overlap the background is less visible because it is seen through two layers of partially transparent “material.”

If we turn on the default, white background in the layers pane we can see how the objects appear against a white background. This illustrates an effect that is occasionally confusing to beginners: the color of the region where both squares overlap is lighter (because of opacity) than the completely opaque square. The logical error made by beginners is to think that if one square has 50% opacity and the other square has 50% opacity then a sight line through both squares should be 100% opaque.

In reality, opacity is more a percentage multiplication effect than addition. When see through one square that is 50% transparent the sight line results in combined color that is 50% from the square and 50% from the background. If we now place another 50% filter of the same color into that sight line the result is color that is 75% the color of the squares and 25% the color of the background. Thus, an overlap of two filters each of which is 50% opaque has the same effect as one filter with 75% opacity and is not the same as 100% opacity.

Layer opacity is a Map Property

Layer opacity is a property of a map layer and not of the drawing or other component that is shown in that layer. The same drawing, for example, can appear in a layer in more than one map and in each map the layer containing the drawing can have a different opacity setting. The opacity used within each map is specified within that map’s layer only.

Changing the opacity of a layer does not make any change to the component shown in that layer. Layer opacity simply changes how the map shows the component in that layer.

Nomenclature

Transparency and opacity are two terms that mean the same concept viewed from different directions. When something is completely opaque it is not at all transparent. When something is perfectly transparent it may be said to have zero percent opacity.
Which word is used depends on the discussion. When imagining layers stacked up above each other like transparent sheets it is conceptually clearer to use the word transparency. When discussing a specific percentage of light transmission to be applied via a slider bar in a dialog most applications use the word opacity.

The convention in the graphics arts editing software industry is to adjust layer opacity with controls that set a number from 0% to 100% opacity, so that an image with 100% will be fully opaque and not allow any view of an image underneath it. Manifold follows this convention. This convention persists in the graphics arts industry even though the technical implementation of transparency effects is done using an alpha channel within RGBa images where the higher the value of the alpha channel (from 0 to 255) the higher the transparency.

One therefore encounters the slight conceptual dissonance of increasing opacity with higher numbers (up to 100%) in dialogs and other user interfaces while the internal data sets use numbers (alpha channel values) in which opacity decreases with higher numbers. Since we rarely set alpha values by hand this is not so bad. Alpha values are normally set using various tools, such as erasers, or masks. In the case of masks, the darker the mask region the lower the alpha value is and thus the higher the opacity. From a casual conceptual view this is very acceptable because it leads to an effect where black regions of masks cause full opacity and white regions of masks cause full transparency. Since we are used to thinking of "white space" as being transparent this works well as a natural mnemonic for the effects of masks.

**Printing and Opacity**

Manifold will correctly print maps (as well as layouts that include maps) with correct opacity for the layers used. However, if the Print using GDI+ option is turned off then only surfaces and images will be printed with correct opacity. Turning off the Print using GDI+ option forces Manifold to use ordinary GDI, which cannot print opacity in drawings.

Although .pdf and .ps formats are often thought of as "universal" formats that can capture exact renderings of data for use in other programs, they do have numerous limitations. For example, neither .pdf nor .ps can render opacity. If a layout includes a map that shows a drawing below an image that has 50% opacity in the map, the .pdf created by exporting the layout to a .pdf file will show the image with 100% opacity. .emf exports do support opacity for images and surfaces but not for drawings.

Although .pdf and .ps formats are often thought of as "universal" formats that can capture exact renderings of data for use in other programs, they do have numerous limitations. For example, neither .ps nor older versions of .pdf can render opacity. If a layout includes a map that shows a drawing below an image that has 50% opacity in the map, the .ps created by exporting the layout to a .ps file will show the image with 100% opacity. Exports using .emf support opacity for images and surfaces but not for drawings.

**Notes**

How are the feathered circles created in the RGBa images shown above? See the Create a Circular Feathered Image example.

Readers outside the US may be surprised to see that congressional districts are not always aligned to boundaries of cities or counties. In fact, they appear to have bizarre, arbitrary shapes in some cases. In the US the political party in power can use its majority vote to draw the boundaries of districts assigned to each congressional representative. This power to draw district boundaries is used to maintain political control by drawing districts based on known demographics of supporters and opponents.

The technical challenge is to draw district boundaries that unite one's supporters and fragment the opposition in a way that prevents opposition voters from ever making up a majority of any district. The practice is known as "gerrymandering" after Massachusetts Governor Eldbridge Gerry (1744-1814) whose second term as Governor featured a bizarre rearrangement of voting districts.
Repositioning Layers

We will often want to move layers up, down, left or right to reposition them relative to each other within a map. Moving a layer within Manifold means repositioning the component that makes up that layer. If the component shows a drawing or image at some geographic location, moving the layer means moving all the objects or pixels in the layer to a different geographic location. This changes the georegistration of the component.

In ordinary graphics editors (like PhotoShop) that work with layers, moving layers is a simple matter of using a move tool to move the layer as desired. Because Manifold always works within a geographic context (even for "non-geographic" images or CAD drawings), moving layers about within Manifold involves some nuances that may require additional thought. There are two approaches to moving a layer within Manifold:

- Repositioning the component using Georegistration or the Register dialog. These dialogs may be used at any time to reposition one component relative to another. Although normally used within geographic maps of various kinds, the georegistration dialog may also be used to align images or CAD drawings in non-geographic work.
- Interactive repositioning using mouse and keyboard moves. These methods may be used when the layer to be repositioned has the same native projection as the projection in use in the map. Since all non-geographic image and CAD work is done in Manifold using an Orthographic projection by default, such methods are available for casual repositioning when using Manifold as a CAD system or as an image editor.

Georegistration and the register dialog are covered in their own topics. This topic discusses interactive methods for repositioning layers.

Requirements for Interactive Repositioning

To move a layer using interactive methods the layer must have the same native projection as is used for the map. The projection in use for a map may be seen in the status bar. The projection used by a particular layer may be seen by right clicking on the layer's tab and choosing Properties from the context menu. To re-project the map window so it uses the native projection of a layer, right click on that layer tab and choose Use Projection from the context menu.

This requirement is automatically met when using Manifold as a CAD editor or as a graphics arts editor. That is because blank images and drawings are created by default in Orthographic projection unless some specific geographic context is specified for them. Images and drawings imported from non-geographic formats (such as those used for graphics arts or photographic work with images or CAD formats such as .dxf) will likewise be imported using Orthographic projection.

The requirement is also met in much geographic work that combines both images and drawings. Images are usually overhead aerial or satellite photos that are best saved in Orthographic projection. It's a good idea to re-project all drawings, images and the maps that show them into the same Orthographic projection centered on the same central latitude and longitude point. This will assure fastest possible performance and will allow use of interactive repositioning methods.

Interactive Repositioning Methods

If the mouse is not occupied in a command mode, we may interactively reposition layers in maps using the following CTRL-Grabber and keyboard commands.

- **CTRL-grabber**: Pressing the CTRL key while clicking and dragging with the Grabber tool will reposition the layer. Using the Grabber tool without CTRL will pan the viewport of the window. Using CTRL grabber will move the actual layer.
- **CTRL <arrow>**: Pressing the CTRL key while pressing one of the keyboard arrow keys will move the layer in that direction.
- **SHIFT - CTRL <arrow>**: Pressing both the SHIFT key and the CTRL key while pressing one of the keyboard arrow keys will move the layer in that direction in greater jumps.
To use these commands, the layer being repositioned must use the same projection as the map. Use the Project to Map choice in the layer tab's context menu to re-project the layer's component to match the projection currently used in the map. This is a permanent alteration in the data.

Example

The illustration shows a map that includes a partially transparent drawing layer above an image layer. The image layer is the SanFran image that has been georegistered to a position close to, but not quite exactly, the desired position. Both the drawing and the SanFran image are in Orthographic projection, as is the map.

We can see that the SanFran image should be moved to the right.

We accomplish this by pressing **SHIFT-CTRL right arrow** a few times and then pressing **SHIFT right arrow** to make the match as exact as possible.

[The illustration shows the city of San Francisco as seen from Landsat 7. The prominent dark rectangular region is Golden Gate Park. The dark region under the triangular point at the top of the San Francisco peninsula is the Presidio, with the point itself being the launching spot of the famous Golden Gate bridge. The bridge itself is just barely discernable between the point and the opposite shoreline.]

See Also

Projections
Orthographic Projection
Georegistration
Manual Georegistration
Merging Layers

Layers can be merged in two ways:

- Copy objects or pixels from one component and paste them into another.
- Create a merged image by using the Tools - Make Image command.

To merge Drawing A into Drawing B:

1. Open Drawing A.
2. Select all objects in Drawing A with Edit - Select All or with CTRL-A.
3. Copy all objects in Drawing A with Edit - Copy or with CTRL-C.
4. Open Drawing B.
5. Paste into Drawing B with Edit - Paste or with CTRL-V. If the data field structure for Drawing A is different than Drawing B, the Paste Objects dialog will be raised to allow us to specify how fields should be pasted.

Do not select any objects in Drawing B before pasting. If any objects are selected in Drawing B just before the paste command, they will be deleted and will be replaced with the objects being pasted.

Copy and paste of pixels also works between images.

To merge multiple image layers into one image:

1. Create a map showing the desired images in the desired layer stack order.
2. Choose Tools - Make Image to create an image component at screen resolution. For greater resolution, first create a print layout of the map, open the layout and use Tools - Make Image to create an image (when used with layouts this command allows choice of resolutions).

A new image will be created in the project that is a combination of the images seen in the map. This image can then be dragged and dropped into a map and used as a layer.

Comments

Why doesn't Manifold simply merge images as does the PhotoShop Merge Layers command? Why require the intermediate Make Image dialog? PhotoShop has a much simpler model of images where pixels are always the same size. Manifold images can be georegistered so that a pixel can represent a very small area on Earth or many square kilometers. The merge process must necessarily resample pixels to interpolate size up or down to merge two images of different size.

Drawings also potentially involve a bit more complexity because two different drawings can have entirely different field structures in their tables.
Maps

Zoom Ranges

The zoom range is a property of projected drawings and other components that appear in maps. Zoom ranges specify the zoom levels at which a component is displayed in a map, including the ability to "pin" the rendering of items to a given zoom level. For zoom levels outside the specified zoom range the component will automatically be hidden from view. The zoom range mechanism therefore makes it possible to create maps where dense components are not seen when the view is zoomed far out and where the component is displayed when zooming far enough into the map.

- Zoom ranges only affect the display of components within maps.
- Zoom ranges only work for projected components. They don't work for Latitude/Longitude unprojected components.

Zoom ranges are set within the native type of component window, but they become operational only when the component is displayed in a map window. Zoom ranges may be set by right-clicking on the layer’s tab in a map window or by opening the layer in its own component window and then choosing View - Properties - Zooms.

To set zoom range, open the component in a window (for example, open a drawing in a drawing window) and choose View - Properties and then click on the [...] button to the right of the Zooms caption to open the Zooms dialog.

The Zooms dialog offers these controls:

- **Minimum zoom**: If blank, show component at any zoom level. If a value is entered, show component at any zoom level above this value and hide it otherwise.
- **Maximum zoom**: If blank, show component at any zoom level. If a value is entered, show component at any zoom level below this value and hide it otherwise.
- **Render zoom**: If blank, redraw points and labels at the same screen size regardless of zoom level. If a value is entered, render points and labels at their specified point sizes only at that zoom level and render points and labels at a larger size when zooming in beyond the specified render zoom level, and render them at a smaller size when zooming out from the specified render zoom level.
- **Current Zoom**: The current zoom of the component when the View - Properties - Zooms dialog was launched. This provides a reference value from which we may get our bearings when deciding what values for zoom ranges or Render zoom we wish to use.
By default the Minimum Zoom and Maximum Zoom ranges are empty, which means the component will be visible in maps at any zoom level. We can choose preset zoom values in factors of ten from 1:1 to 1:1000000000 in the list boxes or we can enter a custom value. The Current zoom number shows the current zoom in the opened drawing window. It will vary as we zoom in or out of the window. It has no effect in setting zoom ranges except to provide a guide that may be used to estimate which zoom values to enter in the Minimum Zoom or Maximum Zoom boxes based on the current appearance of the component window.

![Component window with zoom settings](image)

If we set the Minimum Zoom range value set to 1:10000000 the component will not be visible at all zoom levels up to 1:10000000 after which it will be displayed in maps. If we zoom in to some zoom level below 1:10000000, say, to 1:100000, the component will not be displayed. As we zoom farther out to a zoom level above 1:10000000 the zoom level will appear.

![Component window with custom zoom range](image)

If we like, we can enter a custom zoom range. For example, we can enter 30000000 in the Maximum Zoom box.

This setting of zoom ranges means that the component will be hidden in maps at zoom levels less than 1:10000000 as we zoom farther out and the zoom level goes above 1:10000000 the component will appear. If we continue zooming farther out beyond a zoom level of 1:30000000 the component will disappear. At all zooms between 1:10000000 and 1:30000000 the component will appear.

**Very Important:** Although the zoom range is set when the component window is open, it has an effect only when the component is displayed within maps. Also, zoom ranges can only be set on projected components. They cannot be set for components in Latitude / Longitude coordinates.

See the Turning Layers Off/On by Zoom topic for an example.

**Render Zoom**
The default operation of Manifold within component and map windows is to provide a "dimensionless" rendering of point, line and area styles so that no matter what zoom is selected the styles will appear using the size specified in the format toolbar. For example, if a given point style is specified to be 6 printer's points in size, it will always be shown onscreen exactly 6 points in size. If we zoom in or out the point style will always be shown 6 points in size.

Manifold provides this behavior because vector components such as drawings can be zoomed in or out as much as we like. Normally when, say, many points are clustered together we would like them to continue to be the same size as we zoom further in so that many tightly bunched points can be resolved into visibly separate points.

In the default operation of Manifold, if we wish to have a "paper space" relationship between zooming in and out and the size of symbols such as points, we can use a print layout to create a composition based upon the fixed size of a sheet of paper. When we zoom in and out of a layout, the points and other vector symbols we see will get larger and smaller just as if we were looking at a printed sheet of paper more closely with a magnifying lens or from farther away.

In specialized applications we may want to force component windows to have the same behavior as a print layout display, where the size of labels, points and other symbols will change if we zoom in or out. We can force such behavior by specifying a Render zoom value. The Render zoom value specifies the scale at which styles will be drawn exactly at the size specified in the format toolbar. Zooming farther in, for lower zoom values, will cause the styles to be drawn proportionately larger in size. Zooming farther out, for higher zoom values, will cause the styles to be drawn proportionately smaller in size.

Inexperienced users are often confused between the free-scale world of vector component windows and map windows and the fixed-scale world of a print layout. The normal way of creating displays in which labels and symbols get larger and smaller as we zoom in or out is to use a print layout. Do not use the Render zoom parameter as a sloppy way to get a print layout effect within, say, a drawing or a map window just because the use of print layouts or scaling within print layouts is not understood. Better to study the documentation for print layouts and get those concepts clear so that the right tool may be used.

Keep in mind that one of the significant benefits of component windows like drawings and maps is exactly that they do not mimic the behavior of paper maps. For most purposes the default behavior of component windows provides benefits that would be obscured if symbols and labels were to get larger and smaller as we zoomed in. It is natural for beginners to want to approach GIS using the mental paradigms of paper maps, but it would be a mistake if such an approach denied users the signal benefits of a computerized workspace.

On the other hand, there are times when using a Render zoom value is the right choice. For example, suppose we want to publish a map component to the web using the Internet Map Server and to keep things conceptually simple for some users we want the resultant images to behave like a print layout when they are viewed. In that case, we can use Render zoom to make the image appear like a paper space image so that as visitors zoom farther into the image labels and point styles will appear larger.

**Example**

Consider a map of congressional districts in the US that includes as a layer a drawing containing points for major cities in the US.

We open the map and zoom in to whatever zoom we think will be our "standard" zoom, a typical view.
We then invoke View - Properties - Zooms and note the Current zoom value. It is approximately one to twenty six million.

We can enter 26000000 into the Render zoom box to make this the standard render zoom value.

If we now zoom farther into the drawing the point styles will increase in size. If we zoom twice as far into the drawing the points will be twice as large.
The farther we zoom into the drawing, the larger the point sizes will be, just as if we were viewing a printed sheet of paper through a magnifying lens.

Note that the display computed at the Render zoom value will be the same display, except for increase or decrease in size of symbols and labels, at higher or lower zooms. For example, zooming farther into the map does not allow any overlapping points to be resolved into separate points. This effect can be seen with labels.

Suppose we have a labels layer for the names of major cities in our map and we also set the Render zoom value for that labels layer to 1:26000000 just like the city points layer. Shown above is the map is zoomed to the same starting view, an approximately 1:26000000 zoom. The labels are displayed based upon computations for clipping overlapping labels at that zoom level.

As we zoom farther into the map, the labels simply get larger. No labels are added or deleted.
Just as the case with point styles, the farther we zoom into the map the proportionately larger the labels will become.

Comments

Why are zoom ranges not used in Latitude / Longitude maps? The reason is that such maps use geographic degrees as their unit of measure. Unlike meters or other linear measures used in projected coordinate systems, the size of a degree is different depending on its location on Earth. Degrees near the Equator are sixty nautical miles "wide" (their extent in longitude) but degrees very near the North Pole are only a few meters wide.

Because the size of a degree varies, any scale factor expressed in degrees also varies. If zoom ranges were allowed to apply in maps using Latitude / Longitude, one could have components appear and disappear in maps not as a result of zooming in or out but simply as a result of moving Northward or Southward along a particular meridian of longitude.

See Also

Turning Layers Off/On by Zoom
View - Properties - Zooms
**Match**

The *Match* command appears in the context menu for image and surface layer tabs in maps. The *Match* command matches the image or surface in the active layer to another image or surface within the map. *Match* can be used either with surfaces or images. Any image can be matched to another image or surface. Surfaces can also be matched to another image or surface. To simplify nomenclature, this topic describes its use with images. Usage with surfaces and/or images is exactly the same.

*Match* is used to harmonize two images so that they have the same pixel density, the same height and width and use the same coordinate systems. Using one image as a guide, *Match* will resample, crop or pad with invisible pixels the image being matched so that it is the same size and resolution as the other. All necessary coordinate conversions and maintenance of georegistration will be done automatically. Matching images allows them to be used as masks for each other and to be used together for other purposes.

**To match an image or surface to another**

1. Create a map that includes both components. Verify that they overlap.
2. Right click on the layer tab for the image or surface that is to be altered, and choose *Match* from the context menu.
3. In the *Match* dialog choose the image or surface that is to be matched.
4. Press **OK**.

Because *Match* makes the altered image or surface exactly the same size as the component being matched, after the match process finishes you will only see the uppermost component. Unless the upper component is in a partially transparent layer, the lower image or surface will not be seen because it is exactly the same size and beneath the upper image or surface.

*Match* should be used only with images or surfaces that have some overlap when seen together in a map. See the **Troubleshooting** section below.

**Example**

Suppose we have two images in a map.

One image is a small RGB image called *SanFran* showing part of the San Francisco bay area. This small image was created by georegistering the example *SanFran* image and then cropping it to a small region. This image is in Orthographic projection.
The other image was created by importing the `san_francisco-e.dem` digital elevation module data set from USGS as a surface. The surface was then copied and pasted as an image called `Shading`. Because the original DEM data was in Latitude / Longitude projection the resulting image is also in this projection.

Both images are georegistered, so when we place them together in a map they appear in correct alignment. The illustration shows `SanFran` in a layer above `Shading`. The `SanFran` layer has an opacity of 70% so some of the `Shading` layer shows through from beneath to provide a greater sense of relief. Even though the native projections of both images are different, the map shows them together using the map’s own projection.

We would like to be able to use the pixel data from the `Shading` surface as a mask to alter the values in the `SanFran` pixels. To do this, we need to cut the `Shading` image to exactly the same size as the `SanFran` image. Because the `Shading` image is not in Orthographic projection like the `SanFran` image, it must also be reprojected to match the same coordinate system used by `SanFran`. This will involve interpolation of pixels. Finally, the
image must be re-sampled to be cast into exactly the same number of vertical and horizontal pixels as used by 
SanFran. The Match command performs all of these steps automatically in a single command.

Match operates on the image or surface in the active layer. To match Shading to SanFran, we right click on the 
Shading layer to make it active and choose Match from the context menu. This opens the Match dialog.

The Match dialog lists all of the other images and surfaces in the map. In this example, we only have one other 
image, SanFran. We click that image and choose OK.

The result of the Match is that the Shading image is re-projected into Orthographic projection using the same 
parameters as the SanFran image, georegistered to it, and interpolated and cropped so it is exactly the same 
size, position and pixel density as the SanFran image. Seen in the map window, the Shading image is cropped 
exactly to the size of the other image so it lies completely underneath the SanFran image.

If we open each image in its own window and view the two image windows side by side, we could see how the 
Shading image was cropped.
Padding with Invisible Pixels

In the above example, Match matched the size of the Shading image to SanFran and resampled the Shading image so there is a 1:1 correspondence between pixels in the matched image and those in SanFran. Because the Shading image covered a wider geographic area than SanFran the Shading image had to be cropped.

If the Shading image did not cover all of the region covered by SanFran, invisible pixels would have been added to make up the rectangle covered by SanFran.

Suppose we had used a SanFran image that covered a larger geographic region.
When seen in a map with 50% opacity in the upper SanFran layer, we can see how there is an overlap between the two images and there is also a large region that is covered by SanFran that is not covered by Shading.

We can operate Match as before by right clicking on the Shading layer, choosing Match and then choosing SanFran. The result will be that the Shading image is cropped to the region covered by SanFran. However, in this case it is also extended with invisible pixels.
If we click open the Shading image in its own image window and turn on the Border for that window in the Layers pane we can see that the image extends to the East and North of the gray shading region that was part of the original Shading image.

Matching with Overlaps

Suppose we just want the region of intersection between the SanFran image and the Shading image? There are several possible ways to accomplish this. Using Match we could proceed as follows:

We select the invisible pixels in the Shading image. This can be done using Shift-Touch Select or using the invisible pixels selection in the Selections pane.
Next, we use **Edit - Select Inverse** (or the keyboard shortcut **CTRL-I**) to invert the selection and use **Crop** to crop the **Shading** image to the region shown. This is the region of intersection between **SanFran** and the original **Shading**. There is no change in the visual appearance of **Shading** because we have simply cropped out invisible pixels.

Finally, back in the map window we need to crop **SanFran** to the new **Shading**. To do this we can use **Match** once more. To use **Match**, we right click on the **SanFran** tab in the map and choose **Match**. This time we match **SanFran** to the **Shading** image.

![Map image]

Seen in the map window together, the **SanFran** image with an opacity of 50% in a layer above the **Shading** image, both images are now the same visible size with the same number of pixels.

**Use as Masks**

Once it is matched to **SanFran**, we can use the **Shading** image as a mask to adjust the pixel values directly in the **SanFran** image. It's possible to show **SanFran** in a partially-transparent layer above the **Shading** image map, so the shading shows through as seen above, but this effect cannot be controlled as easily in brightness and saturation and other characteristics as adjusting the pixel values directly in **SanFran** with information from **Shading**.

For this example, we will make a copy of **Shading** that is called **Shading 2**. Because **Shading** is matched to **SanFran** each copy of **Shading** will also be matched to **SanFran**. We use **Brightness / Contrast** to lower the brightness of **Shading 2** so it appears in very dark shades of gray.
Next we click on SanFran and choose Edit - Load Mask / Channel. We load Shading 2 into the blue channel of SanFran using Add mode. This adds the values in Shading 2 to each pixel in the blue channel. We repeat this procedure to load Shading 2 into the red and green channels of SanFran using Add as well. The reason we lowered the brightness of the Shading 2 image before using it as a mask is that we knew by adding it to each of the red, green and blue channels we would increase the overall brightness of the SanFran image.

After some tinkering with Brightness / Contrast with the SanFran image we achieve the image on the left. Note how the relief has been enhanced as compared to the original SanFran image seen on the right. The image on the left has much better mountain relief in the upper left of the image as well as in the lower left and in the mountains along the East Bay in the upper right.

The images used in this example are low resolution, small images. The surface used does not have as striking relief as, say, the Montara Mountain example seen in the Surfaces topics. When surfaces are combined in channels with satellite images in high resolution using dramatic relief the results can be truly breathtaking.

Tech Tips

Why did we make a copy of Shading and use that as the mask? This was done simply so that we would have a copy to work with to alter as we desired, secure in the knowledge that we had the original matched image to use should we like to begin again.

We could have processed the Shading 2 image in many different ways before using it as a mask. For example, we could have used Threshold to force pixel values to black or white for a starker effect. For that matter, we could have used Threshold on the original san_francisco-e surface to render invisible all pixels at sea level. That would have emphasized shading and elevation patterns on land only.

Although this topic uses the example of combining a shading image with a photographic satellite image, we will also use Match to prepare many different other types of images for various channel combinations. For example, we may have one image showing temperatures as seen by a satellite and another derived from a surface showing aspect that we wish to combine. Or, perhaps we have one image that has “large” pixels because it was created
using a satellite instrument with 90-meter resolution that we would like to combine in channels with an image from a different band satellite instrument that sees the Earth in 15-meter resolution. **Match** is a handy tool to help match such images so they can be used together.

**Troubleshooting**

**Images or Surfaces must overlap** - **Match** should be used only with images or surfaces that overlap each other when seen together in a map. It doesn't make sense to use it otherwise. Suppose we had two images in a map that did not overlap and we asked Manifold to **Match** image A to image B. Since no part of image A covers the same region as image B, Manifold will merrily extend image A with enough invisible pixels until image A does cover the same region as B. Manifold would then crop the image to leave only those pixels in locations under image B. In essence, it would result in the creation of an image consisting of invisible pixels only. Such situations can be easily avoided: If the two images don't overlap when displayed together in a map, don't use **Match**.

**Match is not a georegistration tool** - Use Georegistration with control points to georegister one image or surface to another. If the two images or surfaces do not overlap with correct geographic georegistration, **Match** cannot force them to overlap by moving them nor does it perform georegistration.

**Match does not move images** - Use georegistration or the techniques in **Repositioning Layers** to move images into alignment. In the overlap example above, note that those parts of the SanFran and Shading images that do overlap are already in exact geographic alignment.

Think of **Match** as a cropping and re-sampling tool to achieve pixel and size alignment between images that are already georegistered. If large images are being used together in a map it is a good idea for simple system efficiency that the projections used by images and surface be the same, and that this same projection is also the one used by the map. Although **Match** will perform any coordinate transformations necessary to also match coordinate systems (projections) the function will work faster if projection conversion is not also required.

**Performance Note**

**Match** provides comprehensive progress bars so it is possible to see how long a particular match will take and to cancel the match if it is taking too long. Because **Match** performs so many functions in a combined way, large images can take a long time to match depending on the speed of the machine.
Transfer Selection

The Transfer Selection command allows transfer of a selection from one component in a map to another. It is most frequently used to select pixels in an image that are within a selection made in a drawing or vice versa. The Transfer Selection command appears in the context menu when right clicking on a layer tab or in the main menu for that component in the main menu bar. It may be used to transfer selections between drawings, images, surfaces and labels in any combination.

To transfer a selection from layer A to layer B:

1. Create a map that shows layer A and layer B together.
2. Right-click on the layer B tab and choose Transfer Selection.
3. In the Transfer Selection dialog, ensure that the checkbox for layer A is checked and the boxes for other layers (if any) are not checked.
4. Press OK.

The Transfer Selection command will transfer the selection from all checked layers to the target layer specified in the dialog’s Modify box. For example, if an area is selected in a drawing layer and the selection is transferred to an image in the map all pixels in the image that fall within the selected area will also be selected.

When used to transfer selections to objects, the Transfer Selection command will select any object that touches (that is, when any part of the object falls within) an existing selection. Pixels are selected if the center of the pixel falls within a selected object.

Controls

Modify  Transfer the selection to this component. Will be loaded by default with the name of the component whose layer was right-clicked to call up the context menu.

Select All  Use all layers in the map.

Select None  Do not use any layers in the map.

Select Inverse  Check those layers not currently checked and un-check those currently checked.

Using  Transfer the selection in the checked layers to the layer in the Modify box.

Example

This example transfers a selection from objects in a drawing to pixels in an image.
We have a map that contains an image called **Australia Image** and a drawing called **Australia and Oceania** that shows boundaries.

We add a drawing called **Circles** in which we create some circular areas. The areas are shown in blue in the illustration above using a 50% Layer Opacity value.

We select one of the circular areas using touch selection. We would like to transfer this selection to the image so that all pixels that lie within this selected circular area are selected in the image.
We right-click on the **Australia Image** layer tab and choose **Transfer Selection**. In the **Transfer Selection** dialog we check the **Circles** box since we only wish to transfer the selection from that layer and not from the other drawing. Press **OK**.

If we turn off the **Circles** layer in the map we can see that the pixels in the **Australia Image** layer that lay within the selected circle are now selected.

**Example**

The previous example transferred a selection from objects in a drawing to pixels in an image. This example shows the opposite situation, where a selection in pixels is transferred to objects in a drawing.

Suppose we select the pixels of the darker region in the image as seen above. The selection was made quickly using touch selection and **Add to Selection** mode. Note that not all pixels in the region of interest have been selected. For our purposes it will not matter that all pixels in the region of interest have not been selected.
We now wish to select those objects in the Circles drawing layer that touch any of the selected pixels. To do so, we right click on the Circles layer tab and choose Transfer Selection. Note that it does not matter that all of the pixels in the region of interest have not been selected since the size of the circles is such that some selected pixels will be in each of the circles.

In the Transfer Selection dialog we check the Australia Image box to transfer the selection from that layer to the Circles layer.

The result is that all four circular areas that touch any of the selected pixels are selected.

Notes

To transfer selections between components the components must appear together in the same map. If the components are not correctly georegistered and so do not appear correctly aligned in the map then the selection transfer process cannot function.
The examples show transfers between area objects in drawings and pixels in images. Other objects, such as lines and points, can also be used. For example, a frequently used technique to extract information from raster images or surfaces into vector form is to create a fine grid of points in a drawing, select certain pixels in an image and then transfer the selection to the grid of points. The drawing's table can then be opened and a value assigned within some field for all of the selected points.

**Tracing**

Tracing is the process of creating drawings from images. By using the pixels of an image as a guide we can trace over the image and create points, lines and areas in a drawing. The tracing process is like laying a sheet of tracing paper onto a photograph and then drawing lines with a pen to outline what is seen in the photo. The photo image is thus converted into a line drawing. Tracing is also called *vectorization* in older GIS systems.

To trace an image we use a map where the drawing layer in which objects are created appears about the image layer. We can then see both the drawing as well as the image being traced. There are two main methods of tracing within Manifold:

- **Freehand Tracing** At any time we can draw points, lines and areas freehand using the pixels within an image as a guide.

- **Tracing Tools** Tracing tools use pixels in the image layer as a guide to create objects in the drawing layer semi-automatically. For example, the *Trace Point* command will create a point at the center of a pixel cloud.

When tracing an image we will usually use a combination of the above methods. When working with compressed images we can use freehand tracing only, since compressed images do not allow any manipulation as is necessary with tracing tools.

**Example**
Suppose we would like to create a drawing by tracing over a satellite image of San Francisco Bay.

If we insert a new drawing into the project and then insert a map that has the drawing in a layer above the image we can use the Tools toolbar to draw points, lines and areas into the drawing using the image as a guide. The illustration above shows we have added six points and have drawn a magenta line.

If we turn off the image layer in the map we can see just the drawing that has been created.

**Typical Workflow for Tracing**

Most tracing of images in GIS is a five step process:
Acquiring the image.

Georegistering the image.

Preparing the image for better tracing.

Tracing the image.

Editing the resultant drawing.

Acquisition of the image is a key step that often drives what happens in the rest of the process. Scanned images of paper maps are often used in tracing projects. Modern scanners have a bewildering array of options that can result in very different image characteristics for the scanned image. For example, some scanner settings will result in Moiré patterns when printed images are scanned at certain scanner resolution settings. We must learn to operate our scanner software well so that the images we use do not introduce any unnecessary difficulties.

Georegistering the image places it in correct relationship to the location of the drawing and the projections that are being used. This is accomplished via control points referenced to known locations.

Preparing the image involves using Manifold image commands to reduce clutter in the image and to clean up parts of the image that will interfere with tracing. We will often make several copies of the image (we do this after georegistering so all the copies are correctly georegistered as well) so that each copy can be differently processed. Preparation might involve relatively simple procedures such as using Threshold to pick out different colors and make all the rest of the image black or white. We might reduce the number of colors or otherwise simplify the image. We might even apply more involved procedures such as manually editing out text or other items that would interfere with automatic tracing tools. When certain very important parts of the image such as a particular region need to be traced we may take the time to select the region, smooth the selection and then adjust the color within the region to a single color so that it can be automatically traced.

Tracing the image is usually straightforward but time consuming. There is a basic trade-off between quality of tracing and the speed with which tracing is accomplished. Using automated tools is faster but results in lower quality tracing. The highest quality tracing is usually accomplished with the simplest methods: trace points at key intersections and then use those points to create lines and areas in the drawing. Areas are often best created by first tracing lines and then using those lines to build areas within the drawing.

Editing the resultant drawing is an inevitable step in any tracing project. It is often much faster to assume from the very beginning that a substantial amount of editing will be done in the drawing. One can then focus on tracing only those parts of the image necessary to build up the drawing as desired. We will often edit the drawing in one window while leaving the map with both the drawing and the image open in another window. We can then easily switch between editing just in the drawing or tracing and editing with the image as a guide.

Freehand Tracing vs. Automatic Tracing

In this documentation, we refer to manual tracing where the operator draws points, lines and areas as freehand tracing. The Freehand tracing topic provides tips for such work.

Tracing where Manifold creates points, lines and areas is called automatic tracing. The Tracing Tools topic discusses tools available for our use in automatic tracing. Surprisingly, it is often faster and more effective to limit the amount of automatic tracing we would like Manifold to perform.

That humans see what appear to be distinct objects in raster images is a trick of human cognition: there are no objects in images, only patterns of colored pixels within a continuous sea of pixels. Humans learn at a very early age to automatically recognize certain patterns of pixels as distinct things. We can do so even give a lot of extraneous pixel junk and other clutter that our brains automatically filter out.

It is the most natural thing in the world for a human to trace a straight line over a fuzzy, pixelated representation of a line; however, programming this simple human action so that computers can trace a straight line through the same fuzzy pixel cloud has proven remarkably difficult. There are no automatic tracing programs that produce entirely satisfactory results in all cases. At best, certain subtasks can be reliably performed if a human operator participates in the process by giving hints to the computer. We give hints to Manifold on what we way by setting the parameters in the Tool Properties pane for tracing commands.

Part of the difficulty in programming the process is the nature of human choice. All images contain ambiguities. We may think in some cases that it is "obvious" how to trace a particular image but this is usually because we unconsciously assume that "of course" certain choices will be made in a given way. When we trace the image manually we make those choices the way we want to make them without thinking twice.
One key aspect of choice is that images are intrinsically imprecise. At the pixel level images are full of ambiguities and imprecision. A very important part of tracing is making a continuous set of small decisions as to where different features are located and where they start and stop. At times this can be quite arbitrary and is simply a matter of "eyeball" judgement. We know how we like to make those judgements but our computer does not.

Everyone would love a tracing program that could take any image and automatically convert it into a vector drawing, pausing only to read our minds to know what it is we would like to do in the case of any ambiguity. So far there is no such program that comes remotely close to this objective. Most "fully automated" tracing programs are instead used to create an intermediate drawing that is then used as the raw material for manual corrections. Basically, the final drawing is created by manually editing the output of a "fully automated" program.

The Manifold viewpoint is that it is faster to use a semi-automated process right from the start. For these reasons the Manifold tracing tools assume that an alert operator usually will participate in the tracing process. The operator makes the big choices by pointing and clicking and by setting parameters while the computer does the drudge work of averaging pixel clouds to find good centerlines for lines and central locations for points.

See Also

See the Freehand tracing topic for manual tracing and the Tracing Tools topic for use of automatic tracing tools.

Historical Note on Digitizing Tablets

At times users may be curious why Manifold does not support digitizing tablets as a means of converting paper maps or other paper documents into vector data. The reason is that using scanners is much lower cost, much more efficient and much higher accuracy.

Digitizing tablets were flat panels to which a paper map or other document could be attached. A user could move a sensing stylus or puck to different locations on the paper and press a button to note that location (sensing electronics built into the tablet could determine the location of the stylus or puck at the moment the button was pushed). The tablet could then output the various locations so collected, which the associated tablet software could assemble into points or lines as desired. Some tablets provided more sophisticated functions, whereby the tablet could be used as a sort of mechanical user interface (UI) system together with a computer monitor, allowing the user to click the puck on sections of the tablet marked with a command menu.

Digitizing tablets made sense in an era when computers were weak and had very limited capabilities. Before mice and windowing graphical user interfaces (GUIs) became ubiquitous it made sense to cobble up a mechanical interface that allowed point-and-click interfaces no matter how primitive. Digitizing tablets also had the benefit of producing a relatively small stream of data, only the locations clicked, which could fit within the memory and data processing limitations of computers in that era.

In modern times, of course, the Windows metaphor is universal with every computer equipped with a mouse. Scanners have become so inexpensive that less than $50 buys a scanner that can scan at over 1400 DPI in full color. Once an image of a paper map has been scanned into a computer at high resolution, we can then use the full power of the computer to digitize that image as we would like.

An important benefit of using scanned images is accuracy. Humans cannot position a digitizing stylus or puck on paper with accuracy greater than about 10 DPI. However, once an image has been scanned at very high DPI we can then use the power of the computer to zoom into that image and click onto individual pixels if so desired. That gives humans the power to click with accuracy of thousands of DPI, if desired.

Working with scanned images also allows the computer to assist us with tracing tools such as semi-automated tracing, georegistration of image layers, overlays of other layers at the same time and similar assistance.
unavailable with a digitizing tablet. For example, a user wishing to create a new vector drawing of some region for which only paper maps are available might scan in a paper map, georegister that image using control points and then also link in an overhead satellite photo from an image server, displaying the scanned paper map image with partial opacity in a map over the satellite image.

Tracing can then proceed as a freehand tracing interpolation guided by the user's eye between what the paper map shows (which might be inaccurate due to printing or other errors) and what the satellite photo shows. The result can thus be better than using either a scanned paper map or a satellite image alone.
Freehand Tracing

It's easy to create new drawings in Manifold by tracing over images. Although we may choose to new drawings from images using the help of Manifold's tracing tools at times we will simply wish to trace over an image free hand. Freehand tracing works with all image types.

Suppose we have a georegistered image displayed in a map. To trace over it we begin by inserting a blank drawing into the map in a layer above the image. We then use the image as a guide while drawing points, lines or areas over the image.

See the Adding Shapes and the Adding Points, Lines and Areas topics for information on how to draw shapes.

Example

We have created a new map called trace example that was created use the globe sample image. We then created a new blank drawing layer called trace.

From the Tools toolbar we choose Insert Area with the Create Areas option on and the Create Lines and Create Points options off.

In the illustration above, we began by clicking approximately on the Suez Canal and now have clicked our way about halfway around Africa. When drawing areas, there is always a "rubber band" line closing the area from the location of the first click to the location of the current cursor position.
When we click all the way around Africa to the Suez Canal, we right click to finish the command and create the area.

**Notes:**

The created area does not line up very well with the image because we were too far zoomed out for good accuracy when we created the area (we zoomed out to keep the overall image size small enough to fit into this Help documentation). For better detail, we should zoom in further. When inserting a shape, the map view will automatically scroll when the cursor gets near the window border.

The color of the preview line shown in the shapes toolbar can be adjusted in Tools - Options. If we were tracing over a uniformly dark image, the black line shown in the first illustration above would not be visible. We could change this color to bright red or some other color visible on a dark background.

Sensible use of Snap and other guide options will allow us to align our mouse clicks to the underlying image with great accuracy. This example is designed to show the concept of tracing over an image in a simple way.
Tracing Tools

If we do not wish to use freehand tracing, Manifold provides tools to make tracing easier by creating points, lines and areas for us in a semi-automatic way. The tracing tools are found in the **Tracing toolbar**. The Tracing toolbar works within maps to create objects in drawings by using pixels in an image layer to guide the creation of objects in a drawing layer. This function is called **vectorizing** in older GIS packages.

Tracing tools work with all image types except compressed images - to trace compressed images, use freehand tracing.

### Trace Auto

Autoselect between creating areas or points depending on Tool Properties when clicking on a region of pixels of a similar color within the given tolerance.

### Trace Area

Create an area when clicked on a region of pixels of a similar color within the given tolerance. A **Shift click** will create areas within all such regions that qualify as areas throughout the image. Potentially slow in large images or complex shapes.

### Trace Line

Create line when clicked on a region of pixels of a similar color within the given tolerance. A **Shift click** will create lines within all such regions that qualify as lines throughout the image. Potentially slow in large images or complex shapes.

### Trace Point

Create a point when clicked on a region of pixels of a similar color within the given tolerance. A **Shift click** will create points within all such regions that qualify as points throughout the image.

(CTRL click) 

CTRL clicking with a trace tool tells the tool to build objects only in regions where pixels are the same color as the tracing color well. If the tracing tool is clicked on pixels of any other color no tracing will occur.

(SHIFT click) 

SHIFT clicking is an "open" command. It tells the trace tool to build objects in all regions throughout the image where pixels are colored like the place clicked. Caution: in a large image this can take a very long while.

### Pick Color

(Not on the tracing toolbar) This tool is on the editing toolbars for both drawings and images. It is listed here because it is frequently used with the tracing tools. Sample the color in the clicked pixel and use it as the foreground color. A **Shift click** chooses the background color.

Tracing toolbar buttons are enabled when a drawing layer in a map is clicked. Tracing commands require both an image and a drawing layer in the map. The pixels in any visible images layers are used as a guide to create objects in the active drawing layer. Normally we will use several different drawing layers and switch between them so different objects may be created in different drawing layers.

**Tool Properties Controls when Tracing**

All images consist of a sea of pixels. The only thing that differentiates one pixel from another is its color. Tracing tools all work by examine the colors of pixels at the place they are clicked and will continue to search from that initial click point to see how far the clicked color (within the tolerance specified in **Tool Properties**) extends into surrounding pixels. Tracing tools use the parameters specified within Tool Properties to decide how they should deal with the pixels they find that are contiguous with the pixel clicked.

**Snap Tolerance**

When a snap mode is on, the distance in pixels or physical units within which the cursor must be to a given
snap item (such as a line when snapping to lines) before the cursor snaps to that item.

**Value Tolerance**
Color tolerance to use to reckon that pixels are the same color as the pixel originally clicked.

**Point size**
A distance parameter given in native drawing measurement units that is used to distinguish a point from an area. Pixel regions smaller than this distance will be created as points, while those larger than this distance will be created as areas.

**Match neighbors**
When automatically creating areas, fill in the area to align exactly to the edge of any neighboring areas that already exist.

**Trim size**
A distance parameter given in native drawing measurement units that is used to trim unwanted side lines when automatically creating areas.

**Pick Color**

The color picker sets foreground color by sampling a color in an image. Clicking into the foreground color well and drilling down into the color dialogs will show the RGB value of the color chosen with the color picker tool.

Color picker action:

- Clicking anywhere in an image with **Pick Color** chooses the color of the clicked pixel for the foreground color.
- **Shift** clicking anywhere in an image with **Pick Color** chooses the color of the clicked pixel for the background color.
- When the Layers pane is used to turn channels off **Pick Color** will only pick those color values represented by the channels that are visible.
- When only the Alpha channel is turned on in the Layers pane, clicking **Pick Color** into the image will pick a grayscale color value representing the value of the Alpha Channel.
- Clicking **Pick Color** on invisible pixels results in no change.
- When used in a map **Pick Color** chooses color from the highest visible image in the map layer stack.

**Restricting Tool Action by Color**

Tracing tools can be set to use the color shown. Tracing tools normally create objects based on whatever color exists at the point they are clicked. If desired, we can force the tool to use only the specified color by **CTRL clicking** with the tool. This is useful when clicking onto small portions of the image. Set the desired color and then **CTRL click** with a tracing tool to be sure we are clicking only on that color. **CTRL clicking** on other colors will have no effect.

**Tracing Many Objects at Once**

Normally, **tracing** tools will trace just the region of pixels on which the tracing tool is clicked. **SHIFT clicking** on a spot in an image with a tracing tool will cause all regions of pixels of the color clicked to be traced. In Manifold, a **SHIFT** is usually an "open" command. For example, in **Select Touch** a click will select all contiguous pixels like the one touched while a **SHIFT click** with **Select Touch** will select all pixels like the one touched throughout the entire image even if they are not contiguous.

The **SHIFT click** tracing command applies only to that tool used. Using it with **Trace Line** will trace only lines within all pixel regions like the one touched. **Trace Auto** will automatically trace areas, lines and points throughout the image when used with **SHIFT click**.

**Point size and Trim Size**
The **Trace Auto** tool needs to know if it has been clicked on a region of pixels that qualifies as a point or an area. It does this using the **Point size** parameter in the **Tool Properties** pane.

For example, in the illustration above we would want the **Trace Auto** tool to trace the larger purple rectangles as areas and not as points. To make this determination, the tool sees how far the clicked pixel color extends into neighboring pixels and compares it to the **Point size** parameter. If the clicked color extends in all directions no more than the point size parameter, the trace tool knows it has been clicked on a point. If the clicked color extends further than the point width parameter, the trace tool knows it has been clicked on an area.

When autotracing lines the **Trim size** parameter helps the system avoid following a pixel fog into unreasonably complicated lines. If lines are growing undesired side branches, the **Trim size** parameter may be modified.

Note that the **Point size** and **Trim size** parameters are given in the native measurement units of the target drawing. Fractional values are allowed. For projected drawings, the native measurement unit is normally meters or feet. By definition, this means that for sensible use of the **Trace Auto** tracing tool the image and drawing should be in a projection, most likely a meter-based projection.

**Example**

Suppose we wish to trace over some elements of a scanned paper map.

We begin by choosing the **Trace Area** tool and clicking onto the large purple rectangle.

The point where the **Trace Area** tool cursor is first clicked choose the color used to find the area. In this case the tool finds all contiguous purple pixels from the central point. If the **tolerance** factor was set very high the tool would use the lighter purple pixels as well.
The area is created in the region of purple pixels.

We can click on other of the purple rectangles with the Trace Area tool to trace them into areas as well.

If we Shift click on one of the purple rectangles, the Trace Area tool will create areas for all such regions of clicked color.

To create a line we choose the Trace Line tool and then click on the brown contour line just below the two large rectangles we traced into areas.

When we click onto a brown pixel in the line the tracing engine follows contiguous brown pixels up and down to see where the line object should be created.
We show the approximate region where the line is created in magenta pixels, with the line object that was created drawn in black.

On zooming out we can see the line has been created. It is shown as a thick magenta line so it is more easily visible. Note that the new line has been created from the point where the brown pixels ended at the "200" number in the lower left up to the point where the brown pixels ended at the "P" in the "Alpine" letters. While text within the image may be very helpful to readers when it gets in the way of tracing lines it is simply clutter.

We can create points with the tracing tools as well.

To create a point, we choose the Trace Point tool and then click on regions of pixels that we want interpreted as points. In the above case, we can click on the two black dots near the line that was just traced.

The trace tool will follow the black color of the point from the place clicked as shown in the illustration above, which is zoomed far into the image. If our Point Size tool property setting is wider than the geographic distance covered by the pixels that make up the dots they will be interpreted as points and will have a point object created at the center of each.
Because points are often represented by small regions of pixels it is often most convenient to zoom into the point before clicking on it. This avoids accidentally creating a point if we click on some pixel junk near the region on which we intended to click.

**Tracing with Multiple Images Visible**

We may wish to trace using maps where several images are visible at the same time. The tracing tools auto-select which image is used for tracing by choosing the highest visible image at the location where the tool is clicked. That image is then used to guide tracing. This is similar to the action of the **Pick Color** tool, which picks color from the highest visible image in the map layer stack.

The illustration above shows the scanned paper map image used as an example earlier in this topic. Above it in the map layer stack is a drawing and then above the drawing is an RGBA image that contains several semi-transparent rectangles with invisible pixels between the rectangles.

The illustration above depicts the two image layers as we looked at them in a perspective view. The drawing layer (not shown) is between them. The rectangles in the upper image have been made semi-transparent so we can see what happens underneath them.

If we use the **Trace Area** and **Trace Line** tools and click on the points indicated we will be clicking onto the scanned paper map image. In those locations the uppermost image has no visible pixels. Thus, the first visible image layer under the point being clicked is the scanned paper map image's layer.
The result is that the tracing tools create objects in the drawing layer by following pixels in the scanned paper map image. The tracing process follows the pixels in the scanned paper map layer even when those pixels lead "underneath" visible pixels in overlying layers. Note how the traced line object goes underneath the overlying semi-transparent rectangle, for example.

If we would like pixels from several images to guide the action of tracing tools we should first merge those image layers into one image layer (by showing the layers to be combined in one map and then using Tools - Make Image to create a single image) and then use that one image layer to guide tracing.

**Example**

In this example we show how to use tracing tools with surfaces to create areas from a surface. This is an atypical usage of tracing tools that uses an image as an intermediary step.

Let's begin with the Montara Mountain example surface from the Manifold CD. Our objective is to create areas in a drawing that exactly overlay a region of interest in the surface. This is a common task when creating drawings from surfaces, for example, to create vector maps of islands or other features in regions where we have terrain elevation data but no other detail.

We copy the surface and work with the copy of the surface so that any changes made are not permanent.
We begin by selecting the region of interest. We can use whatever selection tools we like, for example, touch selection using add and subtract selection modes.

We press **Edit - Delete** to delete the selected pixels.
We use Tools - Make Image to make an image of the surface, we create a map based on that image and then we open the map.

We zoom into the image to a region that shows the "deleted" part of the surface. Note that when seen in the surface the deleted region shows the default checkerboard background. In the image, which is a literal snapshot of the screen, this part of the image is not transparent pixels but rather consists of light gray and white pixels that render the checkerboard background as it was seen.

We select the region of darker gray boxes by using **SHIFT - Select Touch**.
Next, with selection mode set to **Add to Selection** we select the region of white boxes by again using **SHIFT - Select Touch**.

Finally we select the grid between the boxes with one more **SHIFT - Select Touch** click.
With three quick clicks we have selected the entire region of pixels in the image that represent the region selected and then deleted in the surface.

To make this region stand out we launch the **Image - Adjust - Hue / Saturation** command and adjust **Lightness** all the way to -100. Press **OK**.
This makes all pixels in that region black. This is a color that does not otherwise occur in the image so it is a useful way to mark the desired region with easily discernable pixels.

We add a new drawing to the map and with the drawing being the active layer click on Trace Area and then we click on one of the black pixel regions.
This creates an area in the drawing that overlays the region of black pixels. We click the Trace Area tool cursor into the other black region.

This creates another area overlaying the other region of black pixels.
If we turn off from display the image layer we can see that we have created a drawing with two area objects that precisely overlay the region of pixels that were selected in the original surface.
Labels

Text labels are added to maps with Labels components. To add labels to a map we create a labels component and add it as a layer to the map. We can add the text for labels manually in a labels component, or we can have Manifold automatically create labels by deriving them from objects in a drawing. When labels are automatically created from objects in a drawing, Manifold will use the locations of the objects to place labels and can automatically create the text for those labels by using data from the drawing's table.

A labels component contains information showing where all labels should be placed, the text they contain and how that text should be displayed, including the geometric shape of the label as well as formatting information such as style, font and colors to use.

Labels components may be opened in their own windows or used as layers in a map together with other components such as drawings and images. Except for the labels they contain, labels components are completely transparent and may be stacked as layers in maps to create whatever effect is desired.

For example, the map shown above consists of a drawing layer with two labels layers above it.

The drawing layer shows the shoreline of Europe.
One of the labels layers, called **Land**, is used to host labels indicating regions on land. Labels in this layer are formatted in the style seen above for **Italy** and **Sicily**. Text in this layer is boldfaced using *Times Roman* font in black foreground color.

Although normally layers labels are displayed within map windows, we can also open a labels component in a labels window as seen in the illustration above. Because the labels component is completely transparent except for the text labels it contains, we can see the default checkerboard background in this window. To show a solid background instead of the checkerboard pattern, open the Layers pane and click the **Background** layer on. If the labels component shows a solid background and we prefer the transparent checkerboard pattern, we can click the **Background** layer off in the layers pane.

A second labels layer, called **Sea**, is used to host labels indicating sea areas. Text in this layer is formatted using the *MS Sans Serif* font, boldfaced, in blue foreground color.

The composite effect seen in the map is like stacking two transparent sheets of labels layers above the drawing layer.
We can use many different labels layers to create labels in the map in whatever formatting is desired. The
different layers may be clicked on and off, and can be controlled on a layer-by-layer basis.

Important Options for Labels Components

Manifold may be used in very many different types of applications, so Manifold provides many options for creating
and working with labels. There are capabilities for working with labels manually, for having the system create
labels automatically for us and for a combination of both manual and automatic labels. There are three main
option choices when working with labels:

- **Bound** or **Unbound** labels - Unbound labels are manually created in a free-form way. Bound labels
  are “bound” to a drawing because they are created automatically or semi-automatically from that
drawing. Bound labels may be converted to unbound labels with the Labels - Unlink command at
  any time, but unbound labels cannot be converted to bound labels.

- **Synchronized** or not synchronized - Bound labels can be **synchronized** so they automatically pop
  into existence or are deleted as objects in their parent drawing are created or deleted. Unbound
  labels have no connection to any drawing so this option does not apply to unbound labels.

- **Per-Label Format**, On or Off - By default, the Format toolbar controls the appearance of all labels in
  a labels layer. Turning on per-label formatting makes it possible to independently set the format of
  individual labels. This is very convenient for some purposes but requires extra thought or better
  user organizational skills in some circumstances. Per-label formatting may be turned on or off at
  any time for any labels component using the Labels - Per-Label Format setting.

The choices we make within the above options allow us to specify how labels components behave in our
application, ranging from an exclusively manual, label-by-label control over the placement and appearance of
labels (using unbound labels with per-label formatting turned on), to fully automated creation and placement of
labels (using synchronized bound labels with per-label formatting turned off) or any combination of manual and
semi-automatic functionality in between.

Bound and Unbound Labels Components

Labels components may be **bound** or **unbound**. Unbound labels are manually created, and bound labels may
be automatically created.

- An **unbound** labels component is like a notepad where we can manually add whatever labels we want
  at whatever position we want. Unbound labels components may also be referred to as standalone
  labels components.

- A **bound** labels component is automatically derived from a parent drawing using the positions of
  objects in the parent drawing to specify the positions of corresponding labels. A bound labels
  component is by default synchronized with its parent drawing. By default, there will be a label for every
  object in the drawing and each label will be located exactly at the same position as its parent object.
  Labels bound to lines will by default be aligned to follow those lines, even if the lines are curved lines

Although any labels component may be opened in its own window as well as being used as a layer within a map,
unbound labels components are almost always used only within a map.

COAHUILA

NUEVO LEON

DURANGO

TAMAULIPAS

ZACATECAS

SAN LUIS POTOSI

NAYARIT

AGUASCALIENTES

Because labels components are completely transparent except for the labels they contain, as seen above it is
difficult to maintain visual orientation when an unbound labels component is opened in its own window unless
some other objects can be seen. Unless other layers in a map provide a visual context for position and zoom level it is difficult to work with an unbound labels component on its own.

When a bound labels component is opened in its own window the objects in the parent drawing from which the labels are created will be shown in the background in light gray color as seen in the illustration above. This provides visual context for the labels and makes it easy to work with an unbound labels component either in its own labels component window or as a layer within a map.

A bound labels component may be converted to an unbound labels component by right clicking on the labels component in the project pane and choosing Unlink. The synchronization of a bound labels component to its parent drawing may be turned on or off. The position of label and its metric, that is, the geometric shape that defines that label (normally the location of a point or of a linear shape followed by the label) may be edited.

The Create Labels Dialog

New labels components are created with the Create Labels dialog.

There are three main ways of launching the Create Labels dialog:

- In the main menu, use the File - Create - Labels command. This creates a new labels layer in the project pane.
- Right click in the project pane, either on a drawing component or on an empty portion of the project pane and choose Create - Labels in the pop up context menu. This creates a new labels layer in the project pane.
When a map window is open, right click onto the layers tabs and choose Add - New Labels. This creates a new labels layer in the project pane and after the labels component is created it will be automatically added to the map as a layer.

The Create Labels dialog uses the following controls:

- **Name**: Name to use for the new labels component.
- **Parent**: Contains a list of all drawings in the project as well as a [None] option. Choose the name of a drawing as a Parent when creating a bound labels component. Choose [None] to create an unbound labels component.
- **Columns**: Shows the column names and database type of each column in the parent drawing's table. Enabled when a drawing is chosen as a Parent.
- **Text**: Text to use for the label. May contain a combination of user supplied text as well as the names of columns in [ ] brackets. Column names in brackets will be substituted with the value of that column for the object to which the label is bound. Enabled when a drawing is chosen as a Parent.
- **Automatically label new drawing objects**: If checked, automatically create new labels when new objects are added to the parent drawing. Enabled when a drawing is chosen as a Parent. Checked by default. Labels components with this option checked are said to be synchronized with their parent drawing. See the Synchronized Labels topic.
- **Use per-label format**: If checked, each label will be individually formatted. If not checked, labels will be formatted in accordance with the format toolbar for the labels component until they are selected to be individually formatted. Not checked by default. See the Formatting Individual Labels topic.

To create a labels component we can right click into the project pane and choose Create - Labels. This launches the Create Labels dialog.
If we right-clicked onto a blank portion of the project pane the Create Labels dialog will be launched with no Parent drawing specified by default and those portions of the dialog that would normally work with fields available in a parent drawing will not be enabled. Use this path, or choose [None] in the Parent box to create an unbound labels component.

On the other hand, if we wish to create a bound labels component we can right click onto a drawing in the project pane and then choose Create - Labels in the pop up context menu. In this case the Create Labels dialog will launch with the name of the drawing we right-clicked already loaded into the Parent box.

To Create an Unbound Labels Component

1. Open the map window in which the labels component will be used.
2. Launch the Create Labels dialog, most conveniently by right clicking on a layer tab and choosing Add - New Labels.
3. In the Create Labels dialog, verify [None] is the choice in the Parent box and press OK.
4. Drag and drop the new labels component from the project pane into the map window. This adds the labels component to the map window as a layer. This step is not necessary if we created the labels layer by right clicking on a map layer tab and choosing Add - New Labels, as this method will automatically add the new labels component to the map after it is created.
5. Click on the labels layer tab to make it the active layer.
6. Click the Insert Label button.
7. Click in the map window wherever a label is to be added. In the Insert Label dialog, enter whatever text is desired.
8. Unclick the Insert Label button when finished adding labels.
9. Change the settings in the Format toolbar to change the appearance of the text.

The new unbound labels component will appear in the project pane at its own level and not indented underneath another component.
By opening the map window first and then using File - Create - Labels when the open map window is the active window, we automatically create the new labels component with a native projection that matches the map. Since labels components are almost always used only within map windows it makes sense to create them using a native projection that matches the projection in use in the map window.

To Create a Bound Labels Component

1. Using either File - Create or by right clicking in the project pane and choosing Create - Labels, launch the Create Labels dialog.
2. Choose the name of the parent drawing from the list in the Parent box.
3. Specify the text desired for the label in the Text pane, using fields from the Columns pane above if desired. Press OK.
4. Open the map, and drag and drop the new labels component from the project pane into the map.
5. Click on the labels component's layer tab in the map to make it the active layer and use the Format toolbar to change the appearance of the labels as desired.

The new bound labels component will appear in the project pane indented underneath its parent component.

When creating a bound labels component the labels component will be created using the projection parameters for the parent drawing.

Labels Menu Commands

The Labels menu will appear in the main menu when a labels component is the active component.

- **Add** Appears when a labels layer in a map has the focus. Allows creating and adding a New Drawing, New Image or New Labels component to the map, as well as calling the Add Layers dialog to add or remove layers from existing components in the project pane.
- **Open Drawing** Open the parent drawing for this labels component. Enabled for bound labels only.
- **Open Table** Open the table for the parent drawing of this labels component. Enabled for bound labels only.
- **Text** Open the Text dialog to allow altering label text. Enabled for bound labels only.
- **Per-Label Format** Shows current state of and also toggles per-label formatting on and off for this labels component. See the Formatting Individual Labels topic.
- **Synchronized** Shows current state of and also toggles whether or not this labels component automatically creates a label for each new drawing object. Enabled for bound labels only. See the Synchronized Labels topic.
- **Unlink** Convert a bound labels component into an unbound labels component. Enabled for bound labels only.
Changing Label Text

To change the text of an unbound label, right click on the label and choose Edit from the context menu. In the Edit Label dialog change the text to whatever is desired.

To change the text used in bound labels, in the main menu choose Labels - Text. The Text dialog that is launched reproduces the column pane and the text pane from the Create - Labels dialog so that the text used for bound labels may be changed.

Creating Labels from Fields

Bound labels allow us to create automatically many labels at once by using text fields in tables associated with drawings. We do this by using the Edit - Create - Labels dialog to create a bound labels component that is derived from a particular drawing. See the Creating Labels from Fields topic.

Pasting Objects as Labels

If we copy objects from a drawing and paste them as labels, Manifold opens the Paste Labels dialog to allow us to specify how we want those labels to be created using the following controls:

- **Columns**: Shows the column names and database type of each column in the parent drawing's table.
- **Text**: Text to use for the label. May contain a combination of user supplied text as well as the names of columns in [ ] brackets. Column names in brackets will be substituted with the value of that column for the object to which the label is bound.
- **Link labels to drawing objects**: If checked, create a bound labels component. The default is not checked so that an unbound labels component is created.
- **Automatically label new drawing objects**: If checked, automatically create new labels when new objects are added to the parent drawing. Enabled only if the Link labels to drawing objects option is checked. Labels components with this option checked are said to be synchronized with their parent drawing. See the Synchronized Labels topic.
- **Use per-label format**: If checked, each label will be individually formatted. If not checked, labels will be formatted in accordance with the format toolbar for the labels component until they are selected to be individually formatted. Not checked by default. See the Formatting Individual Labels topic.

The dialog allows us to use content from database fields even if the Link labels to drawing objects option is not checked. If it is not checked, the database field values will be substituted into the label text on a one-time basis to create unbound labels with that text. If it is checked, the database field values will be dynamically applied on a continuing basis to create bound labels using that text.

By default, the Automatically label new drawing objects option is off so that unsynchronized labels are created. This makes it possible to paste a subset of a drawing to create labels only for the pasted objects.

When pasting drawing objects to create labels, the labels component is created using the same projection as the originating drawing.

Labels and Projections

Labels components have projections just like drawings and other components.
• A **bound** labels component always takes its projection from its parent drawing. A bound labels component cannot be re-projected into a projection different from its parent drawing. If the parent drawing is re-projected to a new projection, any labels components bound to that drawing will be re-projected as well into that new projection.

• An **unbound** labels component has its projection defined based upon the circumstances of its creation. If some other component is open and has the focus the unbound labels component will be created using whatever projection that active component uses. If no other component has the focus, then the labels component will be created using the default Orthographic projection. Unbound labels components may be re-projected into whatever new projection is desired.

The above system is very similar to that used for assigning projections when new drawings or other components are created, with the exception that the projection of a bound labels component is always synchronized to the projection of its parent drawing.

**Controlling Zoom Range for Labels**

We can turn labels components on and off for display in maps the same way that drawing layers are turned on and off for display: use Zoom Ranges to set the zoom range over which the labels component should be displayed. Zoom ranges work only with projected labels components and do not work for layers components using a latitude / longitude, unprojected coordinate system.

This capability will often be used when creating labels components that contain lots of labels. The zoom range can be set so labels are visible only when zoomed far enough into the map so that the visible labels can be distinguished from each other. For example, when viewing a map of an entire country a labels layer that shows labels for smaller towns may have zoom ranges specified so that those labels are not displayed when the view is zoomed out so the entire country is seen. The zoom ranges would be arranged so that as we zoom farther into the map so that only more local regions are in view then labels for smaller towns will be displayed. See the Turning Layers Off/On by Zoom example topic.

Note that at some zooms labels may disappear even if a zoom range has not been specified. Labels bound to line objects, for example, will only appear if the line is large enough for the label to fit at the current zoom. As we zoom farther out and lines get smaller, labels bound to those lines will eventually not be able to fit into the smaller space the line occupies on screen.

**Antialiasing**

By default, text labels in Manifold are not antialiased. Checking on the **Antialias text** option in the user interface section of Tools - Options will cause text in labels to be antialiased.

When the **Antialias lines** option is checked (it is on by default) lines, the borders of areas and the borders of text boxes used with labels will be antialiased.

**Creating Line Labels**

In addition to the default **Insert Label** command button shown above, Manifold provides commands to create labels that are aligned to lines.

- **Insert Line Label**  
  Add a text label aligned to the guideline composed of straight line segments that is drawn with the insert tool.

- **Insert Freeform Line Label**  
  Add a text label aligned to the freeform, curved guideline that is drawn with the insert tool.

These tools create labels in the shape of a guideline, either a guideline consisting of straight line segments or a more curved, freeform guideline. The line label tools are much less frequently used than the default **Insert Label** command because the labels they create tend to be useful only at the zoom level at which they were created.

Line labels are aligned to a guideline, which is an invisible line to which the label is aligned. If the label is selected for editing by CTRL-Alt clicking on the label, the guideline will become visible.
Line labels can only use a plain text style or a halo or shadow style. Line labels using other styles will be rendered as plain text. Line labels can only contain one line of text. If more than one line of text is entered when creating a line label the lines will be concatenated together into a single line of text.

**Example**

To use the **Insert Line Label** command, open an unbound labels component and click on the **Insert Line Label** tool.

Click where the guideline is to begin, and then click at the next coordinate of the guideline.

Continue clicking at each coordinate defining each guideline segment.

Click the coordinate for the end of the line and then right click to end construction of the guideline.

The **Insert Label** dialog will pop up. Enter the text desired for the label and press **OK**.

The new line label will appear.
If we unclick the **Insert Line Label** button (so the mouse is not engaged with that command), we can **CTRL-Alt** click the newly-created label to select it for editing. That will cause the guideline to appear with edit boxes at the coordinates defining the guideline. We can adjust the shape of the guideline (and thus, the label) by clicking and dragging edit handles. We can move the entire guideline by **SHIFT**-dragging an edit handle.

Note that the label does not occupy the entire guideline. At the zoom in use the default text size is not large enough to fill the entire guideline.

If we use the format toolbar to increase the size of the font, we can see that the label expands to use more of the guideline.

The size of the label text on the screen will be fixed at the given point size of the font. If we zoom in or out the label text will stay the same size. However, the guideline will appear to be larger or smaller as we zoom in or out because the guideline has a true geographic dimension.

For example, suppose we zoom into the view so that the guideline appears larger. The label text stays the same size and will fill less of the guideline. Because label text is centered on the middle of the guideline, the label text will be aligned to only that much of the guideline that extends from the center to the distance required for the text.
We can zoom even farther in to see this effect.

If we understand how line labels work and we have a label selected for editing so the guideline appears, it is clear how the label automatically aligns to that part of the guideline it occupies at any given zoom level. However, if the label is not selected for editing (the default), it can be disconcerting to see the label change shape as it adapts to an invisible guideline.

For example, if we zoom out the label changes shape...

...and then changes shape yet again as we zoom farther out.

Because of the above behavior, line labels are best used at a given zoom. They are normally used to annotate geographic features where a particular zoom will be used for a final view. For example, we might want to create a curved label that follows the arc of a coastline. The label will have a correct appearance at the zoom level in which it was drawn, but zooming in or out will alter the appearance. That doesn't matter if we are composing a map to be seen at a particular zoom.

**Using the Layers Pane with Labels**

The Layers pane helps control the appearance of labels within label windows. The layers pane includes checkboxes for two system "layers" - a background color layer and a border layer that shows an enclosing box about the height and width of the label.

By default, labels are shown against a white background (or whatever color has been set to be the default background color). Un-checking the **Background** box will use a checkerboard background pattern. Manifold uses to provide a backdrop for any transparent regions. Checking the **Border** box in the layers pane will draw a one-pixel border around the height and width of the label. This is a good way to see the actual extent of a label that contains regions of invisible pixels.

Note that only maps can have true "layers" in Manifold in the sense that they can layer more than one component within the same map window. The border and background "layers" in the Layers pane for labels are not true layers even though they appear in the Layers pane in the same manner as do layers in maps. These are simply system controls that take advantage of the Layers pane as a conceptually convenient user interface.
Layouts and the Layers Pane

If a labels component has any Layouts created they will appear as "layers" in the layers pane for that labels component. Checking the box for one of these print layout layers will cause a layout rectangle to appear in the labels window that shows the region covered by the layout.

Right clicking onto the hatched border of one of the layout rectangles in the labels window will cause a context menu to appear with controls based on that layout rectangle. For example, we can Zoom to a given layout rectangle, Print it or change its Properties. If a layout is empty (for example, if the layout scope is set to selection and no labels are selected in the parent labels component) zooming to the layout will do nothing.

Use Tools - Options - Colors - Layout Rectangle to change the color in which layout rectangles are shown. The default color is black.

Advice for New Users

Users new to GIS who are familiar with graphics arts editing packages such as Adobe Illustrator are sometimes baffled by the focus of Manifold on automated facilities for the construction of labels. There are several reasons why GIS has evolved a more automated focus on label creation and manipulation than have graphics arts packages:

- GIS is used for much more than static cartography, so it must have capabilities to support those other uses.
- GIS displays are normally automatically-created, dynamic displays while graphics arts packages are used to create static, predictable compositions.
- GIS tends to work with immensely larger data sets than do graphics arts packages.

Graphics arts professionals new to GIS often do not realize that the usage of GIS packages to create static cartography, that is, to create fixed maps for publication, is but one small part of how GIS packages are used, however important that task may be. If your job has been creating maps with a graphics arts package such as Adobe Illustrator, it is only natural that your first exposure to GIS would arise from such tasks and that your first interest would be in how to use a GIS package to accomplish similar cartographic objectives.

However, one reason graphics arts professionals are drawn to GIS is that GIS provides many capabilities, principally those revolving around the tying of database information to geography and geometry, not found in graphics arts packages. The main value of GIS in most applications is the use of visual representations of data, such as maps, to get a handle on complex data relationships that would be very difficult to see by scanning through rows and columns of letters and numbers.

Labels play an important role in many such uses, but just as numbers must be automatically computed in spreadsheets for the spreadsheet to work, in GIS applications using visual displays to explore data, labels must pop into existence as needed in the form needed to support such data management.

In these and many other uses of GIS the visual display of data in GIS packages is highly dynamic while the purpose of a graphics arts package like Illustrator is to create a single, static composition.

Automatic creation and control of label characteristics is very important because many times the projects we create in Manifold will be seen through views that are automatically created and that cannot be predicted in advance. For example, if we create a map with many layers in Manifold and that map is displayed to people through a Manifold Internet Map Server web site, each view seen in that web site will be automatically created depending on how that particular user zooms or pans into the map. In such views labels had better be automatically created, sized and placed so that each view that is created is reasonably attractive and useful for its intended purpose.

Even in the simple case of ordinary interactive usage of Manifold, when someone pops a map open on their desktop and then zooms in or out or pans the map to different locations the views will constantly change in terms of zoom level or location. Any labels that are in play must automatically be adapted to the new view for the labels to play their intended role. They must be created or hidden, moved to slightly different locations, change their shape to adapt to a different shape of a line at different zoom level and so on.

That's a situation very different from Illustrator or, say, PhotoShop, where we might spend a lot of time getting some graphic image exactly right. But every professional graphics designer who has spent hours of effort on an
Manifold® System Release 8.00 User Manual

Illustration for a journal article or for an advertisement has done so based on a specific, static understanding of the composition. If the map changed to show different material the composition would also have to change.

Finally, the amounts of data encountered in GIS are vastly greater than are used with graphics arts packages. Much GIS data is automatically harvested by satellites and then converted into huge raster and vector data sets, possibly with connections to hundreds of fields of information for each object in the map. A data set representing a world map might contain hundreds of gigabytes of information and even a simple local map might include many thousands of objects and tens of megabytes of data. A GIS working with such data must have the ability to manage large amounts of data to reduce it down to what is of interest.

Labels created from such data are best created automatically, since few people are interested in manually sifting through tens of thousands of objects to decide which should be labeled by hand. In fact, automation of such tasks is exactly why a graphics arts professional might be forced to undergo the supremely unpleasant task of learning a new and unfamiliar software package.

It is true of course that none of the above points prevent a GIS package from acquiring the facilities for manual graphics arts editing that we take for granted in a package like Illustrator. However, the above factors place the priority of what features are implemented by the GIS vendor more on automated label facilities to support the distinct usages of a GIS package. As time goes on and Manifold continues to evolve, more and more features similar to those in graphics arts packages will be available in addition to those distinctly GIS features already implemented.

Users who have strong feelings about desired labels features should visit the Support page on the manifold.net web site and read the link on that page that gives advice on making suggestions and user advocacy and then send in a suggestion for what is desired.

**Tech Tips**

Labels components may be copied and pasted as a drawing. The result will be a drawing of points where each point has a "Text" field in the drawing's table that contains the contents of the label text. Copy labels and Paste As a drawing is used in two situations:

- When importing some formats, such as .e00, the result of the import may be a labels component where we would also like to have a drawing showing a point at each label location.
- Labels components may be exported to AutoCAD DXF format. See the Export Drawing - DXF topic.
- Copying a labels component and pasting as a drawing is a quick way to move the labels text into a drawing of points that have one text field. Exporting such a drawing might be a useful way of exporting the labels component to a different format. The labels text can be a composite text made up of several different fields, so this is a quick way to capture that text into a single field.
- For convenience, right clicking on a drawing in the project pane and choosing Create - Labels will launch the Create Labels dialog with that drawing already highlighted in the dialog's Parent pane under the presumption that we would like to create labels for that drawing. To not pre-select a drawing in this way simply right click into the blank region of the project pane when choosing Create - Labels from the context menu, or click anywhere in the blank region of the Create Labels dialog so that no drawing is chosen.
- By default, Manifold will autoclip labels (not show some labels) to prevent the display from being too crowded with labels. If you add labels and they disappear or some other labels disappear, they are being autoclipped by the overlap resolution algorithm. To turn off autoclipping open the labels component in its own window, choose View - Display Options and un-check the Resolve overlaps box.
- When using a mouse equipped with a scroll wheel, the scroll wheel zooms in and out of a labels component. Holding the CTRL key down while scrolling will force the zoom to operate near the center of the window.
- Manifold provides one-step Undo when deleting or duplicating a label or when editing a label metric.

**See Also**

Creating Labels from Fields
Synchronized Labels
Formatting Labels
Formatting Individual Labels
Thematic Formatting and Labels
Label Display Options
Creating Labels from Fields

Labels that are created from fields are always bound labels in Manifold. Bound labels are automatically created from objects in a drawing, including possibly fields taken from the drawing's table, so they are "bound" to that drawing and its objects.

Bound labels may also be synchronized with the drawing as well. If the labels component is synchronized with the drawing, then adding new objects to the drawing will create new labels for those objects and deleting objects from the drawing will delete their associated labels. See the Synchronized Labels topic for more details.

Whether or not bound labels are synchronized, if their text is taken from fields then they will automatically be updated with any changes in the drawing's data fields. For example, changing the formatting of the table's columns, such as changing a numeric field to show two digits after a decimal point, will automatically change the format of that numeric field as it appears in a bound label as well.

Creating a labels component that prints text based on data fields is the same as creating any bound labels component. We simply use the names of fields within square [ ] brackets in the Text pane of the Create Labels dialog.

There are several main ways to create a bound labels component:

- In the project pane or from the File menu choose Create - Labels and then choose a Parent drawing from which labels will be created using fields from that drawing's table.
- Copy either an entire drawing or some or all objects from a drawing and then Paste As a labels component. This will raise the Paste Labels dialog to create a labels component that is bound to the originating drawing.
- Right click in the project pane, either on a drawing component or on an empty portion of the project pane and choose Create - Labels in the pop up context menu. Choose a Parent drawing in the resulting dialog. This creates a new labels layer in the project pane.
- When a map window is open, right click onto the layers tabs and choose Add - New Labels. Choose a Parent drawing in the resulting dialog. This creates a new labels layer in the project pane and after the labels component is created it will be automatically added to the map as a layer.

Whichever path is used to raise the Create Labels dialog, we can illustrate the process using the File - Create - Labels command as follows:

To create labels using fields in drawings:

1. Choose File - Create - Labels
2. In the Create Labels dialog, click on one of the drawings listed in the Parent combo list.
3. This enables a list of fields in that drawing in the Columns pane. Double click on a field and it will appear in the Text pane.
4. Edit the Text pane as desired.
5. Press OK

Double-clicking a field name to place it within the editing pane is a useful shortcut but is not necessary. If desired, we can enter field names such as [Place_name] manually. Double clicking is recommended to avoid typographical errors in the names of fields.

When a labels component containing bound labels is opened in a labels window the parent drawing will be displayed in gray default color as a background to provide a visual context for the position of the labels.

Example

We will use the Mexico example drawing to create labels.

![Mexico Drawing](image1.png)
![Mexico Table](image2.png)
We begin by importing the drawing into the project. We then right click onto the **Mexico Drawing** component in the project pane and choose **Create - Labels** to launch the **Create Labels** dialog.

The dialog shows **Mexico Drawing** loaded into the **Parent** combo box since we right clicked onto that drawing. If we had right clicked onto an empty portion of the project pane or if we had launched the **Create - Labels** dialog by choosing from the **File** main menu then the **Parent** box would have been loaded with **[None]** and we would have had to choose **Mexico Drawing** from the list made available in the **Parent** combo box.

If we had other drawings in our project these would be available in the **Parent** list as well. When we choose a drawing in the **Parent** box the columns for that drawing will appear in the **Columns** pane.

In the **Columns** pane we scroll down to **Place_name** and double click it. Double clicking a field name in the **Columns** pane will load it into the text in the **Text** pane. We could also simply enter the field name in brackets in the **Text** pane; however, double clicking the name from the **Columns** pane is often faster and less subject to typographical error.
We can now click into the *Text* pane and add any desired literal text. In this example, we have added "*My name is*” to the text for the label. When the label is constructed the field names in brackets will be replaced by the value in that column for that label's parent object. All text not in brackets is taken as literal text and will be displayed exactly as entered.

We can now click into the *Text* pane and add any desired literal text. In this example, we have added "*My name is*” to the text for the label. When the label is constructed the field names in brackets will be replaced by the value in that column for that label's parent object. All text not in brackets is taken as literal text and will be displayed exactly as entered.

**Note that intrinsic fields will also be listed in the **Columns** pane. If desired, we could scroll down to and double click on the **Latitude (I)** and **Longitude (I)** fields to add these in brackets to the **Text** pane. Intrinsic fields are computed by the system for each object. The **Latitude (I)** and **Longitude (I)** intrinsic fields report the centroid position of an object in the map. See the Intrinsic Fields in Tables topic for more information on intrinsic fields. When we are done adding text in the **Text** pane we press **OK**.
A new labels component appears in the project. Since it is created using fields from Mexico Drawing it appears in the project hierarchy under its parent component.

We can create a map that shows Mexico Drawing and the Labels label component together as layers with the labels layer above the drawing layer. Note how values have been substituted for [Place_name], [Latitude (I)] and [Longitude (I)]. The label is positioned at the centroid of the area. Each area in the drawing has a label.

Changing Formatting of Numbers in Labels

Suppose we would like to create labels with fewer digits after the decimal point in the latitude and longitude values? The formats of numbers in labels are taken from the format used for the columns in the table that provide those numbers. To change the appearance of numbers in labels that are taken from columns, change the formatting of those columns in the table. See the Formatting Columns topic.

For example, to change the appearance of the latitude and longitude intrinsic values, open the table and display the intrinsic columns. Next, change their formatting by right clicking on their column heads and choosing Format. Changing the format of the column will also change the format of any label that uses numbers from that column.

Changing Text used in Bound labels

If we would like to change the text or fields used in bound labels like the example above, open the labels component in its own window and choose Labels - Text to launch the Text dialog.

Creating Labels for Some, but not All, Objects

At times we may wish to create labels for some, but not all, objects in a drawing. We have several methods available:

- Select those objects (such as areas in a drawing) for which labels are to be created. Copy the objects and paste them as a drawing, and then create labels for that drawing.
- Un-check the Automatically label new drawing objects option in the Create Labels dialog, so that the resultant labels component is not synchronized. Alternately, after the component is created un-check Labels - Synchronized from the main menu so that it is not synchronized. We can then select and delete all labels we don’t want. If only a few out of many labels are desired, it will be faster to select those labels we do want, choose Edit - Select Inverse to invert the selection and then delete. See the Synchronized Labels topic for more info.
- Create a field that can be used for thematic formatting so labels that are not to be displayed can be thematically formatted using invisible color and desired labels can be formatted using a visible color. This technique allows only some of the labels to be made visible.
- Create unbound labels manually. The disadvantage of this method is that label text cannot be taken automatically from fields as is the case with bound labels. See the Adding Text Labels topic.
- Create bound labels and then use the Unlink command to convert the bound labels component into a standalone, unbound labels component. The labels may then be repositioned or individually deleted or altered.
- Use linked drawings to create a drawing from a query where the results of the query are the drawing objects we want to have labeled. Create a synchronized, bound labels components using that dynamic drawing as the parent and then use that labels component. This can be faster than the thematic formatting approach because the labels component only contains those labels desired.

Example

In this example we create labels for some provinces in Mexico, using thematic formatting to show only some labels but not others.
Consider a map with one layer, the example drawing of Mexican provinces. We add a labels layer by right clicking on the **Mexico Drawing** layer tab and choosing **Add - New Labels**.

In the **Create Labels** dialog we choose the **Mexico Drawing** component as the parent and double click the **Place_name** column to add it to the label text. Press **OK**.
This creates a new labels component, called **Labels**, and also adds it as a layer to the map. Note that labels are created for all provinces. The default auto-clipping will hide some labels that would otherwise overlap other labels. Overlap resolution may be adjusted using the **View - Display Options** dialog for the labels component.

At this point, the smart thing would be to simply un-check **Labels - Synchronized** to make this an unsynchronized labels component. We could then simply select and delete unwanted labels.

However, since this is an example for Help that is intended to illustrate various ways of working with fields in tables in connection with labels, we will use a trickier approach, that of thematic formatting. This technique does not make much sense in ordinary interactive usage, but it is often used in IMS applications where we may want labels to appear or disappear based upon the contents of data fields.

We will create a new data column and the use that to guide whether or not a label is visible.

In the project pane we double click on the table for the drawing to open the table.

We right click on any of the table’s column headings and choose **Add - Column** to add a column to the table.
In the Add Column dialog we create a new Boolean field called Label. A Boolean field can have only two values, which depending on the formatting for the column in use, can be represented as 1 or 0 (default formatting), True or False, Yes or No, On or Off and so on. Boolean fields are handy when a value may be either on or off.

There is no special meaning to our choice of the name Label for the new column. It is only a mnemonic to remind us that this field's values will be used to turn labels on or off for display.

<table>
<thead>
<tr>
<th>STVE</th>
<th>Place_name</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>3199</td>
<td>SÁN LUIS POTOSÍ</td>
<td>0</td>
</tr>
<tr>
<td>3537</td>
<td>Zacatecas</td>
<td>0</td>
</tr>
<tr>
<td>1505</td>
<td>Durango</td>
<td>0</td>
</tr>
<tr>
<td>3398</td>
<td>Sonora</td>
<td>0</td>
</tr>
<tr>
<td>5780</td>
<td>Sinaloa</td>
<td>0</td>
</tr>
<tr>
<td>3234</td>
<td>Tamaulipas</td>
<td>0</td>
</tr>
<tr>
<td>3200</td>
<td>Nuevo León</td>
<td>0</td>
</tr>
<tr>
<td>3536</td>
<td>Baja California</td>
<td>0</td>
</tr>
</tbody>
</table>

The new field is created with default 0 values. Our task now is to fill the Label cell with a 1 for those provinces for which we wish to have labels displayed.

This is easily done by clicking on the drawing layer in the map to make it the active layer and then selecting those provinces for which we want labels. We use touch selection in Add to Selection mode to click each province to be selected.
If we switch back to the drawing's table, we see that those records corresponding to selected provinces are also selected. We double click into the **Label** cell for any of the selected records...

...enter a 1 and then...

...when we press **Enter** the new value of 1 will be propagated into the **Label** value for all of the selected records. Manifold uses auto-scoping for selection editing in tables, so that a change made to any selected record is a change made to all selected records. This is a very convenient way of modifying many records at once.

We have now marked all selected provinces with a value of 1 in their **Label** column. We can use this column to thematically format the labels component so that labels with a 1 in their object's **Label** column are visible and those with a 0 in their **Label** column are invisible.
Click on the labels layer in the map to make it the active layer and then in the pull-down menu for the **Foreground Color** well in the formatting toolbar choose **Theme**.

This launches the thematic formatting dialog. We choose **Label** as the field to control the formatting. We double-click into the **Values** color well for the 0 value and choose invisible color. We will continue to use the default black color for the 1 value. Press **OK**.

**Note:** the thematic formatting dialog shows both 0 and 1 values in the **Values** pane for our editing pleasure because there are already some 0 values and also some 1 values in the table. If we did not previously change some of the values to 1 in the table the formatting dialog would only show 0 value in the **Values** pane.
In the map window we can see that now only labels for selected states will be visible, because other labels are rendered in invisible color and cannot be seen. Not all labels are seen because some (such as the label for Durango province to the East of Sinaloa) will be suppressed by the auto-clipping algorithms designed to prevent overlapping label text.

We can clean up the display by deselecting the selected provinces in the drawing. We can also change the formatting of labels in the labels layer to turn on **Bold** face so the labels stand out better.

The example above works well to display labels for only a subset of objects in a drawing. However, it has the performance impact of computing labels for all of the objects in a drawing and then reckoning which of those labels should be displayed with invisible color (so they cannot be seen) and which should be displayed for visible color. If we have 100,000 objects and only wish to show labels for five of those objects the above technique is somewhat wasteful because it requires the computation of 100,000 labels even though only five of the labels will be seen.
A more efficient technique would be to create a drawing consisting only of the five objects for which labels should be displayed and then create a labels component bound to that drawing. In that case, only five labels would be in play in the labels component. We could create such a drawing dynamically by using a query to create a linked drawing based upon the other drawing and then create a labels component that is bound to the linked drawing.

**Showing Multiple Labels per Object**

Sometimes we would like to show more than one label for objects. For example, we might have a drawing showing cities as points with population and name fields. We would like to show a label for population and a label for the name for each point.

![Text: Name, Population]

One way to accomplish this is to create a single label that uses the contents from both the name and population field as seen above.

Another way to show two labels for each point we create two labels components. We copy the cities drawing and paste it once as a labels component using the name field. We then paste it a second time as a labels component using the population field. Use a different alignment for the two labels components so that the two labels do not appear on top of each other. We then use both labels components in a map with the drawing. When using this method it may be important to turn off the **Resolve overlaps between labels** option in the **View - Display Options** dialog for labels so the different label layers do not clip each other.

**Labels without Fields**

At times we may want to create labels for a bound component that do not use fields.

Consider a drawing that shows oil well locations as points.
We create a labels component using this drawing as the parent and use the text "Oil Well" without using the names of any fields in brackets.

We can drag and drop this labels component into the same map as the Oil Wells drawing to see that it creates the same text for every object. We have used the alignment toolbar to move the text labels slightly above and to the right of each parent point and we have used the Display Options dialog to add three pixels of space via the X and Y offsets between the labels and the points as well.
Let's now create another labels component for this drawing but this time we will use a field to customize each label with the contents of the Production data field.

We can drag and drop the new labels field into the map to create another labels layer. This labels component also has the labels moved slightly and uses the format toolbar to boldface each label.

**Tech Tip: Using Bracket [ ] Characters in Labels**

Given that bracket characters [ ] are used in the Create Labels text pane to indicate a field name from which data should be taken, how can we create text that includes a bracket character? Manifold at the present time does not allow an escape sequence, such as \[ to be used to denote the next character should be taken as a literal. Instead, we must use a hack.

Create two new text columns in the drawing's table. Call one LeftBracket or similar name and call the other RightBracket or similar name. The text columns may be created as fixed-length text with a length of 1 character. Select all records in the table and double-click into the LeftBracket column and enter a [ character. Because Manifold offers the convenient shortcut of filling all selected records with the edited value this will populate the LeftBracket column with a [ character for all records. Do the same for the RightBracket column except populate it with a ] character.
Now, whenever we want to use a left bracket in labels text we simply use the expression `[LeftBracket]` and whenever we want to use a right bracket we use the expression `[RightBracket]`. For example, if we want to create a text label like `Verified [1995]` we would enter the following into the text pane:

```
Verified [LeftBracket]1995[RightBracket]
```

The contents of the "bracket" field names will be taken from the field which in all cases will be either a left bracket character or a right bracket character as desired.

Note that we can combine this technique with other field names as well. Suppose we have a `Date Verified` column in the table that has the particular year each record was verified, so that instead of hard-coding 1995 as the year we want to take the value from the field. We could do that by writing:

```
Verified [LeftBracket][Date Verified][RightBracket]
```

**Note:** It's quite likely at some future point Manifold will add an escape sequence allowing direct entry of bracket characters without the need for the above technique. However, the above technique should still be kept in mind as there may always be special characters in various fonts that can only be accessed through programmatic or other special means. In that case, an active column can be created that generates them and then the name of the active column can be used within brackets in the label text.

**A Note for Enterprise Edition Users**

It is possible to create a local labels component that is bound to a shared drawing component. This allows creation of a labels component within our project that is created based upon a drawing linked in from an Enterprise server.

**See Also**

Labels
Formatting Labels
Formatting Individual Labels
Thematic Formatting and Labels
Label Display Options
Synchronized Labels

When creating bound labels we get a choice in the Create - Labels dialog to **Automatically label new drawing objects** or not. If this option is checked (the default) the labels component will be **synchronized** to the parent drawing.

When labels are synchronized to their parent drawing, if any new objects are created in the parent drawing then new labels will automatically be created for those new objects. If any objects are deleted from the parent drawing then their associated labels will also be deleted from the parent drawing. When labels are synchronized to the parent drawing we cannot delete any labels manually since their appearance in the labels component is based exclusively upon whether or not their associated object in the parent drawing exists.

When labels are not synchronized to their parent drawing, we may freely delete or (based upon objects) add labels to the labels component. We can select a subset of labels and then copy them to a new labels component (which will also be an unsynchronized bound labels component). We can add objects to the parent drawing without adding any labels, but if we delete any objects from the drawing their labels will also be deleted. That is because even an unsynchronized label is still a bound label and derives its existence from the associated object in the parent drawing. We can choose whether or not the label is shown but as long as it is bound to a parent object we cannot have the label exist on its own without the parent object that defines it.

Because labels in a bound drawing are always derived from the parent drawing in any event, choosing synchronization on or off is really a choice of how labels are created and whether or not they are deleted. It is not a choice as to the content of labels or their position.

**Example**

Let's begin with the sample **Mexico** drawing used throughout this documentation. This drawing has a **Place_name** field we can use to automatically create bound labels.

Using the **Create Labels** dialog we create labels from the **Mexico** drawing using the **Place_name** field. We use the default setting (checked) for the **Automatically label new drawing objects** option. We will call this new labels component by the default name, **Labels**.
We use the default setting (checked) for the **Automatically label new drawing objects** option. The resulting labels component will therefore be synchronized to the parent drawing.

A new labels component appears in the project pane, indented below its parent drawing because it is bound to that drawing.

We can add the resulting labels component to a map that also contains the Mexico drawing. In the above illustration the labels have been formatted so they are slightly larger than default and also so they use a style with a "drop shadow" of white background color.

With the labels component layer tab clicked in the map so that the labels layer is the active layer, we can click on the **Labels** menu to see that this labels component is **Synchronized**. It is synchronized because the labels component was created with the **Automatically label new drawing objects** option checked on.

If we desired, we could un-check the **Synchronized** setting in the **Labels** menu to turn off synchronization for this component. For now, we will leave it synchronized so we can see how a synchronized labels component behaves.
The first thing we might notice is that synchronized labels cannot be deleted. We can select all of the labels in the labels component with a **CTRL-A** (the standard Windows power user "Select All" shortcut) or by using **Edit - Select All**. If we now press the **Delete** key, nothing happens, nor is the **Delete** command enabled in the **Edit** menu.

Let's now create a similar bound labels component, but let's create this as an unsynchronized labels component.

Once again we use the **Create Labels** dialog to create labels from the **Mexico** drawing using the **Place_name** field. This time, however, we will **un-check** the **automatically label new drawing objects** option so the result is an unsynchronized labels component. We will give this new labels component the name **Unsynchronized Labels** so that we can tell at a glance in the project pane which labels component is which.

A second labels component appears in the project pane, also indented below its parent drawing because it too is bound to that drawing.

We click off the synchronized labels layer in the map so it is not visible and then we drag and drop the new labels component into the map. We will leave these labels formatted using default colors and style so it is clear in these illustrations when a synchronized labels layer is shown and when an unsynchronized labels layer is illustrated.
If we click on the **Labels** menu we can see that it reports this labels component is not synchronized: there is no check mark next to the **Synchronized** option.

Let us now try some experiments to see how unsynchronized labels behave.

We will first select two labels near the center of the view and press **CTRL-X** to cut them (or, press **Edit - Cut**).

Both labels will be deleted and copied to the Windows Clipboard. Note that it is not only possible to delete labels in an unsynchronized labels component; it is also possible to just delete some of them.

If we like, we can right click into the project pane and **Paste** to create a new labels component using the content of the Windows clipboard.

A third labels component appears in the project pane, named using the default naming scheme Manifold applies when new components are pasted. This labels component is also indented below its parent drawing because it too is bound to that drawing.

When we copy labels from a bound labels component and paste them into a project, the new labels component will inherit the bound status of the component from which the labels were copied. The pasted component will be bound to the same parent drawing, it will use the same text (including use of any fields) and it will use the same projection.
If we open the new labels component in its own window, we can see that it has two labels in it, the two labels that were cut from the original and pasted to create this labels component.

Let's go back to the unsynchronized labels layer in the map and select all of the labels.

If we press **Delete**, they will all be deleted. We can now add labels back to this labels layer with the assistance of a highly useful shortcut.

**Using the Insert Label Tool with Unsynchronized Labels Components**

The **Insert Label** command is used to insert a label. If we click with this tool into an **unbound** labels component it will pop open a dialog allowing us to enter text for that label. In a **synchronized** labels component this tool is not available, because synchronized labels always are created or disappear based on whether their parent objects in the drawing are created or disappear.
However, in the case of an unsynchronized bound labels component, the **Insert Label** tool is available. When clicked into the labels component, it will create a label at the position clicked for the object at that location. The text of the label will be taken from whatever is the current specification for labels in that bound labels component.

To continue the example, we can open our unsynchronized labels component in its own window and click into it with the **Insert Label** tool. Each time we click at a location a label will be created at that spot for the object at that spot.

For example, if we click at the location above, within the area representing **Zacatecas** province...

...a label will appear there using the **Place_name** field for that object. If we click at the next location...

...a label will appear for **San Luis Potosí** province. We can click lower and to the right...
...and a label will appear for Jalisco province. If we click in the region of Durango province to the West of Zacatecas...

...a label will appear there as well.

This technique is a very rapid way of creating labels just where we want them. We might have many objects, such as all of the provinces in a country, but want to label only some of them. A fast way of accomplishing that goal is to create a bound labels component that is not synchronized, delete all the many labels created automatically (easy to do with a CTRL-A and a CTRL-C using the Windows keyboard shortcuts for Select All and Delete) and then use the Insert Label tool to click at desired locations to create labels only where desired.

The labels will also appear, of course, within the map as well since we have used the same labels component as a layer in that map.

The technique above was illustrated using a labels window, but it works for a labels layer in a map window as well, so long as the map includes as layers both the parent drawing and the labels component bound to that drawing and the parent drawing layer is visible.
Let us now see what happens to bound labels when we delete their parent objects.

We will select the area representing Zacatecas and press Delete.

This deletes the area from the drawing. It also deletes the associated label from all labels components bound to that drawing. Note that there is no longer a label for Zacatecas. Even though we are looking at an unsynchronized labels layer, it is still bound to the parent drawing and the labels cannot exist without their parent objects.

Let us now see what happens when we add an area. First, we will turn on Instant Data mode so we can add data attributes quickly to newly-created objects.

Next, we will use the Insert Area tool to add a triangular area in the empty region formerly occupied by Zacatecas province.
When we add the new area, Instant Data mode will pop open the Object Fields dialog to allow us to add values for this new object. We will add the value Manifoldia for the Place_name of this newly-created area. "Manifoldia" strikes us as a nice name for a new province named after Manifold System.

In the unsynchronized labels layer, no new labels appear. That's because with unsynchronized labels adding new objects will not create new labels for them.

However, we can click off the unsynchronized labels layer and click on the synchronized labels layer to see what has happened there.
In the synchronized labels layer a new label has appeared for Manifoldia. That’s because with synchronized labels any new object added to the parent drawing will automatically have a new label created for it.

**Multiple Labels per Object**

Bound labels components can have more than one label for every drawing object, and the metric of each such label may be independently labeled (see the Editing Labels topic for information on editing label metrics).

There are two main ways in which an object in a drawing can have multiple labels bound to it:

- More than one labels component can be bound to the same drawing, so objects in the drawing can have labels bound to them that are in more than one different labels component.
- An individual labels component can have more than one label bound to the same drawing.

The first situation above is fairly obvious, since we can easily create more than one labels component bound to the same drawing.

It is less obvious how we could create multiple labels bound to the same drawing object within the same labels component. This is best illustrated with a small example using a bound, unsynchronized labels component as follows.

Consider a bound, unsynchronized labels component we’ve created for the sample Mexico drawing just as in the example above. The illustration shows the labels component open in its own window.

After deleting all of the labels created when the component was first created, we’ve used the Insert Label tool to create a few labels. The illustration above shows the window just as we are about to click the tool into the Northwest corner of Durango province to create a label there.
Manifold creates a label for us at the position clicked using the data for Place_name taken from the object that was clicked. Now we will click the Insert Label tool again near the center of Durango.

Once again, a label is created at the clicked spot. We can click a third time in the Southwest portion of the province...

...and a third label is created. What is going on is that we have created three new labels, all of which are bound to the area object representing Durango province.

We can see this by clicking into the drawing’s table and changing the Place_name value for this object.

If we change DURANGO in upper case to title case, Durango, the change will be immediately reflected in all three labels bound to that object.
The above technique allows us to add multiple labels per object. There are many circumstances in which we might want to have multiple labels per object.

For example, we might want to place labels near the borders of a large country or province, as seen above. The illustration (a zoomed in view of Durango) was created by turning on per-label formatting for this labels component and then selecting each label in turn and using individual rotation format values for each of the three labels.

Another common usage for having multiple labels per object is to create more than one label for a long line, such as a highway, that should be labeled at more than one location.

Notes

The above example uses the Insert Label tool. The other label inserting tools also work in this way, including the Insert Line Label and Insert Freeform Line Label tools.

Although the above example is purely interactive, we should keep in mind that using synchronized labels can be extremely useful in non-interactive settings as well when queries or other processes are used to dynamically create drawings. For example, suppose a linked drawing is automatically created from a query or from an external DBMS table. It is very useful to have any labels associated with such a drawing be automatically created using a synchronized labels component since we don't know in advance how many objects will be in that drawing or where they might be located. Such automatic creation of labels is extremely important in dynamic applications, such as Internet Map Server applications.

See Also

Labels
Creating Labels from Fields
Formatting Labels
Formatting Individual Labels
Thematic Formatting and Labels
Label Display Options
Formatting Labels

By default, all of the text labels in a labels component are formatted the same way based on the settings of the Format Toolbar for that labels component. See the Formatting Individual Labels topic for information on specifying formatting for individual labels.

Labels may be formatted using static settings as shown in the format toolbar, or they may be dynamically formatted in size, style, foreground color, border color, background color or rotation using thematic formatting as set forth in the Thematic Formatting and Labels topic.

At any time, we can change the formatting of any labels component by simply clicking on its layer tab in a map window or by clicking it open in its own labels window and changing the settings in the format toolbar for that labels component.

Consider, for example, the illustration shown above, which shows the Sea labels using the MS Sans Serif fault.

We can click open the Sea labels component and change the font to Monotype Corsiva with a font size of 14.
The result changes not only in any labels window for the Sea labels component but instantly in the map window as well.

At any time we can click on any labels layer tab in the map window to make it the active layer.

We can then change the formatting toolbar for that labels component. In this case we’ve changed the font to ScalaSans - Caps (a commercial font designed by Martin Majoor in Utrecht and available at www.fontshop.com) in 18 point size in "outlined" style.

Note: for the "outlined" style to work correctly the Antialias text option must be checked in Tools - Options - User Interface.

The result is an instant change in the label formatting for the Land labels layer.

Label Colors

The default label style uses foreground color only. Some label styles will embellish the label through the use of border color and background color as well. In most styles using both foreground and background color the foreground color is overlaid upon the background color. Some styles will combine the foreground and background colors to create gradients.
**Label Styles**

Different styles are used for point labels and for line labels. Line labels use text only because they can be aligned to their host line. Point labels can use styles such as boxes and highway shields.

**Example**

Consider a map of Europe over which we have created an unbound labels component. Using the **Insert Label** tool we have added a label with the text "Europe."

![Map of Europe with label](image)

The label appears in the default format.

![Default label format](image)

The format toolbar shows default settings for font characteristics such as size and the use of default colors for label foreground, border and background and default style as well.

![Font size change](image)

To change the size of the font from 8 points to 20 points we click on the font size well in the format toolbar and we select 20 from the pull-down menu of available font sizes.

![New font size](image)

The format toolbar shows the new font size and the labels component will use that new font size for labels.
Next we will change the label style to a style that draws labels in a box.

For example, we can change the **Label Background** color to white.

This will change the color of the "fill" color in the box in this style.
We can change the **Label Border** color to red.

![Label Border](image)

Instantly, the border of the label's box changes to red.

![Label Foreground](image)

Likewise, we can change the **Label Foreground** color to blue...

![Label Foreground](image)

...and the change will immediately occur in the labels display.

**Transparent Color**

Transparent color, shown with an **X** in the color well, may be used for any color attribute.

![Transparent Color](image)

To specify transparent color as the background color we click on the background color well and choose transparent color.
This makes the white background fill color disappear from the label box, so that layers below the labels layer may be visible through the box.

Using transparent color for foreground color will make the entire label disappear. This is a short-hand convenience for turning labels off via thematic formatting.

Consider a label formatted like the above. The toolbar shows white foreground, red border and black background colors to create the display shown.

If we change foreground color to transparent color the entire label disappears.

If we use any color other than transparent color for foreground color the entire label reappears.
This short-hand convenience uses foreground color as the controlling color because it doesn't make much sense in most situations to use transparent color for the foreground color of a label since foreground color is normally used for the text of a label, the salient aspect of most labels. If transparent color were to be used for the text, the text could not be seen and the label would consist of whatever background, if any, is used by the style in use.

**Enterprise Edition**

Sharing a labels component with per-label formatting to an Enterprise server will save the necessary formatting information within that component. Importing or linking a shared labels component with per-label formatting will read the necessary formatting data from the server along with the component.

**Note**

The `/slist` command line option provides a handy way of getting a list of all available label styles and their names for use by programmers. See the Command Line Options topic.

**See Also**

Labels
Formatting Individual Labels
Thematic Formatting and Labels
Label Display Options
Formatting Individual Labels

By default, the format settings used in the Format toolbar for labels apply to all labels in a labels component. If we change a format setting, all labels in that labels component will be changed.

There are two ways to individually format labels within the same labels component:

- For bound labels components only, we may use **Thematic Formatting** to change the format of labels based upon the values of data fields in their parent objects. See the Thematic Formatting and Labels topic for more information on this method.
- For both bound and standalone labels components, we may use **Per-Label Formatting** by turning on the **Per-Label Format** option and manually specifying the format for any labels we want to be different from the overall format specified for the component.

This topic deals with the latter method, the use of per-label formatting.

To individually format a label:

1. Open the labels component, or click on a labels layer in a map.
2. In the **Labels** menu, make sure the **Per-Label Format** option is checked.
3. Choose settings in the Format toolbar that will apply to most labels in the component. All labels will be formatted with these values.
4. Using any Selection method, select one or more labels to be individually formatted.
5. Change the settings in the Format toolbar to the desired format for the selected labels. These labels will now be uncoupled from the overall settings for the component and will retain their selection setting until they are reset to the default or until they again are selected and formatted differently.

When a labels component has **Per-Label Format** turned on, there may be two types of labels in the component:

- Those labels that use the **default format** specified by the toolbar for default use in the component. If there is no selection made, changing settings in the format toolbar when per-label formatting is turned on for a labels component will change the default format for that component. All labels using the default format will be changed.
- Those labels using a **custom format** that have been selected and re-formatted while they were selected. When labels are selected, changing settings in the format toolbar will change the custom format for those labels.

To change a label from using custom formatting back to using the default format, select the label and then use the **Edit - Reset Format** command to reset that label back into default format.

Adding a new label to a labels component assigns that label the default format.

**Example**

Let's look at per-label formatting in action. We begin by creating a bound labels component from the **Mexico** sample drawing using the **Place_name** field to create labels of place names. We open that component in its own window.
The labels component opens showing default formatting for the labels.

In the Edit menu we make sure the Per-Label Format option is checked.

If we make a change in the format toolbar, such as by changing the default Tahoma font to Times New Roman...

...it will be applied to all labels in the component.
To apply a custom format to some subset of labels we first select the desired labels. In the illustration above three labels have been selected using **visual selection** with the **Select Box** tool.

If we now change the Format toolbar, such as by pressing the **Bold** button to boldface text...

...the new format will be applied only to selected labels, which are now shown in boldfaced text. These labels have now been detached from default formatting and will be formatted only according to their custom format.

We can see this by deselecting the three labels and making a change in the format toolbar.
For example, we can change the font back to **Tahoma**.

Note that now the change in the default format does not affect the three labels which have been detached from the custom format and assigned their own custom formats. All of the other labels now use the default **Tahoma** text format but the three labels assigned custom formats have not changed.

At any time we can see what custom format a label uses by selecting it. The format toolbar will show the settings used by that label. For example, in the previous two illustrations we have selected the **Nayarit** label and can see from the format toolbar it uses boldfaced **Times New Roman** font.

If we would like to reconnect a label to the default format we can select it and choose **Edit - Reset Format**.
This will immediately restore whatever is the default format for the labels component to that label and re-attach it to the default format.

By selecting labels and then applying different formats we can create an immense range of different label formats within the same labels component.

**Custom Label Formats do not Use Thematic Formatting**

Thematic formatting can only be applied to the default format. If a label is detached from the default format and assigned a custom format, that custom format cannot be a thematic format.

It is possible to select some labels and then specify a thematic format by choosing **Theme** in the format toolbar's sample wells. However, when the thematic format is applied to one or more labels using a custom format the correct format for each particular label will be computed on a one-time basis and applied to that label. Thereafter when the label is selected the format toolbar will not report a thematic format for that label but will simply report the settings that were computed and applied on a one-time basis, as if they had been manually selected and applied.

**Tips**

Use per-label formatting with restraint. It is annoying to viewers if too many different label formats are in use within the same visual display. Usually it is best to stick to a handful of label formats.

Do not use per-label formatting as a lazy way to avoid taking the time to organize labels into subsets that are displayed in different labels components, which are then stacked up in a map to achieve the desired composite effect. It is much quicker to change the format toolbar once and thus change the formatting of all labels in a layer than it is to hunt down and change by hand each label's custom format.

In a large, complex project it can be easy to forget that a particular label was once formatted with a custom format and is now no longer attached to the default format. One technique to find labels that are not attached to the default format is to deselect all labels and then choose transparent color for the label foreground color. This will
Labels

cause all labels attached to the default format to disappear. We can then select all of the remaining labels using a visual selection method like Select Box to thus select all labels that use a custom format.

Avoid using transparent color for foreground color when per-label formatting is enabled and any labels are selected. If we select a label and then change foreground color to transparent color, the entire label will disappear. This will make it difficult to select that label using the usual visual selection tools like Select Box, since visual selection tools don’t work on invisible labels.

If this happens, don’t panic. Zoom out so that all labels are in view. The label may be retrieved by first hitting CTRL-A to select all labels (CTRL-A selects all labels whether they are visible or not). Next, change the Selection Mode to Select Subtract and then use Select Box to draw a selection box covering all visible labels. This will deselect them because selection mode is set to subtract. What will be left in the selection after all the visible labels have been subtracted will be any invisible labels. We can then change foreground color in the format toolbar to some visible color and the invisible labels will pop back into view.

Keep in mind if things get completely out of hand and we lose track of what formats have been applied to what labels it is always possible to start over by forcing all labels in a given component to be attached to the default format: Select all labels with a CTRL-A and then choose Edit - Reset Format.

Because selected objects are shown by default in red selection color it can be difficult at times to imagine how a new format will appear when the red selection color does not obscure the colors chosen in the format. That can interfere with our ability to choose pleasing color schemes. We can avoid this problem by using None for Selection Style in the selection toolbar so that selected objects are not redrawn in red color while they are selected. This does require paying closer attention to which labels are selected since red selection color will no longer make them stand out; however, usually the changes in formatting make it obvious which objects have been selected and are now being re-formatted.

Changing the formatting of labels can move labels about and cause some labels to disappear because Display Options to resolve overlaps or to optimize horizontal or vertical positioning are active. These options may be clicked off if their result is annoying or otherwise unwanted.

See Also

Labels
Formatting Labels
Thematic Formatting and Labels
Label Display Options
Thematic Formatting and Labels

When bound labels are created from fields in a drawing, the labels may be thematically formatted based upon the values in the fields. The foreground color, border color, background color, size, style and rotation of a label may be controlled by a field.

If per-label formatting is turned on the thematic format specified will apply to all labels in the labels component using the default format. If per-label formatting is turned on, thematic formatting is not dynamically applied to labels using custom formats.

To thematically format a labels component:

1. Open the labels component, or click on a labels layer in a map.
2. In the Format toolbar, click on the display characteristic to be formatted. Choose the foreground color, border color background color, style or size. Rotation may also be controlled by a field.
3. In the pull-down choice menu that appears, choose Theme.
4. In the thematic Format dialog, choose the controlling Field to be used.
5. Choose the Method to be used to construct intervals. Equal Count, for example, will assign approximately the same number of objects to each interval.
6. Choose the number of Breaks between intervals. This specifies the number of intervals as well.
7. Change the Align to value if even numbers of tens, hundreds, thousands, etc, values are desired for intervals. Change the Range if the default entire range needs to be extended or contracted.
8. Press the Tally button to create the given number of intervals. If interval numbers different than those created by the Method are desired, double-click into the interval numbers to change them.
9. Either use a preset color palette or click into the interval color boxes in the Values pane to change colors to whatever range is desired. To use a palette, select it in the Palette box and press Apply to apply the preset to the intervals. Check the Preview box to try out different combinations and see how they look in the labels component without committing the changes.
10. When satisfied, press OK.

See the Thematic Formatting topic for a detailed discussion of controls in the Format dialog used for thematic formatting.

Example

Suppose we have a bound labels component created for a parent drawing of cities in the US that has a Name field and a Population field for each city. Suppose the labels were created from the drawing using the Name field using the procedure set forth in the Creating Labels from Fields topic.

In the illustration above we see the labels in a layer above a background map. The view is zoomed into a region within the great state of Texas. We would like to change the formatting of each label based upon the Population field. We'll begin by changing the size, so that labels for cities with larger population will be shown in a larger font.

In the format toolbar, click on the size button.
In the pull-down menu that appears, choose Theme.

Within the Format dialog that appears, choose the Population field as the controlling field. We'll use the Equal Count method so that there will be approximately the same number of labels of each size. To round off the interval values, we'll choose 4 as the number of digits for Align to and then choose Tally to recompute intervals to even 10,000 values.

On the right side of the Values pane are two columns of numbers. The column of numbers more to the left shows the number of objects in each interval and the column more to the right shows the current value of font size for each interval. Note that the rightmost number for each interval is 8, the default font size. By double clicking on each font size value we can change it.
For example, here we have changed the font sizes from all 8's to a range from 6 to 14. Press OK.

The result is that labels for cities with larger population will be shown in larger font sizes.

We can also change the label style, if desired. To do so, click on the label style button in the format toolbar and choose Theme.
Once again, we will choose Population as the controlling field and Equal Count as the method. We will also Align to 4 digits and press Tally to round off the intervals to the nearest 10,000.

We will change the style for only one interval, the largest, by double clicking on the style sample and choosing one with a drop shadow effect in background color. Press OK.

The result is that the largest labels are now shown in a drop shadow style.

To illustrate a change in foreground color, we can click on the foreground color button in the format toolbar and choose Theme.
As before, we will choose Population as the controlling field and Equal Count as the method. We will also Align to 4 digits and press Tally to round off the intervals to the nearest 10,000. We will choose the Spectrum palette and apply it to the intervals to color them automatically.

The result is that labels are thematically colored based upon the size of the controlling Population field for each label. In this particular case the effect is not very sophisticated given the garishness of the palette used, but it does illustrate the process of thematically formatting label foreground color.

Note that in the above example we used the same field and the same method with the same number of intervals to format size, style and foreground color. It is very easy to use thematic formatting to vary too many variables at once and so create a confusing presentation. See the comments in the Thematic Formatting Example topic for advice on avoiding overly complex, and thus incomprehensible, presentations.

See Also

Thematic Formatting
Rotating Labels
Zoom Ranges
Turning Layers Off/On by Zoom

Format Toolbar
View - Properties - Zooms
Aligning Labels

Manifold provides a number of commands to quickly align labels relative to their default positions. These may be used to position labels to the upper right of points they identify or other relative positions. Alignment commands are in the Format toolbar for labels.

The default positions for labels depend on the method used for their creation:

- **Unbound** labels will be created at the spot the label creation tool was clicked to create that label. The tie points of unbound labels may be moved about and the alignment of the labels relative to their tie points may be changed as well.
- **Bound** labels are automatically created at the inner centroid of an area, are placed at the midpoint of a line and are centered on the position of a point. The tie points of bound labels may not be moved about, but the alignment of the labels relative to their tie points may be changed.

Tie Points and Tie Lines

Tie points and tie lines are the point locations and line shapes to which point labels that define the location and shapes of point and line labels.

Point labels are positioned within the labels component based on **tie points**. In the case of unbound labels, the tie point is the spot clicked when we use the **Insert Label** command to insert a new label manually.

When bound labels are created automatically using data fields, the tie point is the inner centroid for the object. When bound labels are created using points the location of the point will be the same as the location of the tie point.

We can see the tie point by opening the labels component in a window and choosing the View - Structure - Tie Points command.

Label Text

If we have two labels we can see their tie points by checking on tie points.

The tie points appear as small gray boxes. We can move the labels by selecting them for editing and then moving the edit handles that appear at the tie point locations.

Line labels are drawn aligned to **tie lines** which may be seen by choosing the View - Structure - Tie Lines command.

Consider a line label that was created in a labels component by using the **Insert Line Label** command.
If we turn on **View - Structure - Tie Lines** we can see the virtual line to which the line label is anchored. Note in the above illustration that a fine gray line now appears showing the tie line to which the label is anchored.

We may wonder why viewing tie lines is at all necessary, since selecting any given label for editing with a CTRL-ALT click will show the tie line to which it is anchored as a side effect of showing edit handles.

The main reason is that turning on **View - Structure - Tie Lines** turns on tie lines for all labels in a labels component window at once, saving us from having to select each in turn for editing just to see its tie line.

Viewing tie lines for labels can be very useful in circumstances where the text specified for the label is not long enough to fill out the line at the given zoom. For example, the text in the above line label does not fill out the entire line.

By turning on tie lines we can see exactly what is the relationship between the text and the tie line defined for the label.

**Object Boxes**

Object boxes are rectangular boxes that outline the labels text. They are guides that appear only in a labels component window when the **View - Structure - Object Boxes** menu item is checked. They are used to see the alignment of labels text relative to tie points.

Suppose we have a text label that consists of multiple lines. [We will use a lengthy text label in this example because it shows the object box better.]

> Open the book to the first page and begin scanning words from left to right until their meaning is understood.

If we enable **Object Boxes** we see the outline that encloses the text.
If we enable Tie Points we see the tie point to which the text is bound and relative to which it is positioned.

The Align controls in the Format toolbar adjust the position of the label object box relative to the tie point for the label.

- We can click the Align Left button to position the text object box to the left of the tie point.
- We can click the Center Horizontally button to center the text object box on the tie point. This is the default position.
- We can click the Align Right button to position the text object box to the right of the tie point.

**Align Controls**

- **Align Left**
- **Center Horizontally**
- **Align Right**
The left-center-right and top-center-bottom buttons can be combined, for example, to create a right-bottom alignment.

**Justification**

Manifold label text justification applies to all labels in a given labels component using the *Left*, *Center* and *Right* justification buttons in the format toolbar.

For example, clicking the *Left*, *Center* and *Right* justification buttons in the format toolbar arranges labels as follows:

Open the book to the first page and begin scanning words from left to right until their meaning is understood.

For obvious reasons, justification effects are best seen with multiline labels.

**Aligning Labels to Lines**

Bound labels created automatically from fields in line objects are always aligned automatically to their parent lines.

**Note**

The text in the multiline label examples above is taken from the "How to Use This Book" section of *Mr. Bunny's Guide to ActiveX*, by Carlton Egremont III, ISBN 0-201-48536-2. One of the funniest programming texts ever written, this slim volume presents ActiveX concepts in the form of conversations between a cartoon rabbit and farmer.
Label Display Options
The View - Display Options dialog may be launched when a labels component has the focus. The dialog allows us to specify how labels will be displayed or clipped from display. The unit of measurement employed is the printer's point unit used in printing.

- **Point Labels**: Point labels are those used for points and for areas.
  - **Offset X ... points**: Distance in points in X (horizontal) displacement from the anchor point of the label. Increase distance to position labels farther away from the anchor point and to accentuate the effect of label alignment settings. 0 by default. Not used if alignment buttons are set to place the label directly upon the point.
  - **Offset Y ... points**: Distance in points in Y (vertical) displacement from the anchor point of the label. Increase distance to position labels farther away from the anchor point and to accentuate the effect of label alignment settings. 0 by default. Not used if alignment buttons are set to place the label directly upon the point.

- **Callouts**: Draw a line from the label to each point from which the label is generated.
- **Minimum length**: When a callout line is used, do not show it unless it is at least this long.
- **Optimize horizontal alignment**: Alter the horizontal alignment of labels to improve the positioning of labels.
- **Optimize vertical alignment**: Alter the vertical alignment of labels to improve the positioning of labels.

- **Line Labels**: Line labels are those used for lines.
  - **Offset ... points**: Distance in points that a line label should be offset from a line. 0 points by default.

- **Character Spacing**: Extra distance between characters in the line label text.
  - **Label each branch**: If checked (default), will place a line label on each branch of a multi-branched line. If not checked, will place a line label only on the longest branch.
  - **Multiple labels per branch**: If a particular line's branch is long enough, label it with multiple, repeating labels. This is useful for long lines.
  - **Spacing**: The minimum spacing between multiple labels on the same branch of a line.
  - **Orient labels left-to-right**: If checked (default), Manifold orients each label bound to a line to read from left to right as shown on the screen. When the option is turned off, Manifold orients each label so it flows from the beginning of the line to the end of the line.
  - **Render labels near short lines**: If not checked (default), labels bound to lines that are too short to fit the label text will be hidden. If checked, labels bound to short lines will be rendered as if they were bound to the center of the line. If a line is multi-branched the center of the first branch will be used.
  - **Resolve overlaps between labels**: If checked (default), automatically hide some labels so that labels do not overlap.
  - **Spacing**: Space allowed between labels in points before overlap resolution, when enabled, will take effect. Default is 1
Examples

With Offset for lines of 0 points the line labels appear on their parent lines.

Increasing Offset to 6 points moves the labels off the lines.

When labels are aligned to lines using the default Offset of 6 points they will be positioned six printers' points away from their line.

If we increase the Offset parameter to 24 points the labels will be positioned farther away.
Increasing **Character spacing** to 3 points adds extra space between the letters in the label text.

The **Offset X** and **Offset Y** parameters for point labels works in a similar fashion to the **Offset** parameters for lines. The default distance is seen above, using **Align right** and **Align top** alignment.

Changing the **Offset** parameter to 5 points in both **X** and **Y** directions moves the labels farther away from the point.

**Optimizing Label Alignment**

Labels are normally aligned as directed by the alignment toolbar mode buttons. Checking the **Optimize horizontal alignment** or the **Optimize vertical alignment** boxes in the display options dialog allows Manifold to override the alignment settings for any given label to improve the display.

For example, in the above illustration the rightmost point does not have a label because with current alignment settings the label is automatically clipped because it would conflict with other labels. In the above illustrations, the **Optimize** alignment boxes are not checked.

If we check the boxes, then (as seen above) Manifold can alter the alignment of the rightmost label so that it can appear immediately to the left of the point and thus not conflict with other labels.

**Orienting Line Labels**
The **Orient labels left-to-right** option allows us to orient labels to either follow the direction of the line in all circumstances or to always read left-to-right regardless of the direction of the line.

To understand this option we must understand how lines are internally constructed within Manifold. Consider the line above. A line inside Manifold is defined by a sequence of internal coordinates between which segments of the line are drawn in a connect-the-dots fashion.

We can see these coordinates by **CTRL-ALT** clicking the line to select it for editing. An edit box will be drawn at each coordinate that defines the line. Coordinates for lines are not stored at random within Manifold: each line consists of a list of coordinates from beginning to end, with the segments of a line drawn from each coordinate to the next coordinate in the list.

This means that there is always an implied **directionality** to any line in Manifold, in that each line begins at a first coordinate and then continues through the list of coordinates to a last coordinate. The first coordinate is the beginning of the line and the last coordinate is the end of the line. The line may be said to run from the first coordinate to the end.

We cannot see the direction of the line in the above illustration. Although it is obvious which two edit handles are at the end of the line, we don't know which is the first one and which is the last one in the sequence. That is, we don't know if the line was drawing beginning with the coordinate on the left and then continued to end at the coordinate at the right, or if the line was drawing beginning at the right and ending at the left.

Because English text as well as that of most other languages is read from left to right there is a great temptation to always assume lines have been drawn from left to right but that is not always the case, since we have no idea how lines in a drawing were created. They may have been created, for example, by automated processes that have drawn lines so that directionality ends up being random.

The orientation of lines is important because label text can follow the directionality of lines. However, when we print labels regardless of the directionality of the line we would like those labels to read left to right as much as possible. The **Orient labels left-to-right** option (which is checked by default) allows us to be sure that will happen. Let's see how this works:

Consider a drawing consisting of three lines with a labels layer containing three bound labels shown above it in a map. By default, we can't tell what is the directionality of any of the lines.
We can, however, show the directionality of each line by choosing an arrow line style for the drawing that shows lines with an arrowhead at the end of each line. Immediately, we can see that two of the lines have directionality more or less left-to-right while the red, selected line has directionality right-to-left.

In a simple drawing like this, which was probably created by hand, what probably occurred was that the person who created the drawing began at the central point and then created the two lines at the right by clicking the line creation tool starting at the central point. The third line was created by clicking at the central point and then clicking off to the left so the directionality of the third line ended up being right-to-left.

The default setting for the Orient labels left-to-right option causes the Base Camp Trail label to appear right-side-up regardless of the directionality of the selected line. If we un-check the Orient labels left-to-right option in View - Display Options we cause the letters of the label to follow the directionality of the line as if the line were rotated about a clock face to its current position.

Troubleshooting Labels Placement

Manifold creates one line label per line. If multiple labels are being created for what appears to be the same line, it is most likely that what appears to be one line is really composed of multiple line objects placed exactly end to end. Verify this by selecting one of the lines to see if only part of the "line" appears in red selection color. Multiple lines can be joined together into one line with the Dissolve command.

Disappearing Labels

By default, labels bound to lines will only appear if the line is long enough to fit the label text. As we zoom out from a given view and lines become shorter and shorter their labels will start disappearing as the lengths of the lines become too small to fit the label text. We can change this behavior by checking the Render labels near short lines option. When checked, this option shows labels even if the lines to which they are bound are too short to fit the label text.

Non-GDI+ Printing Limitation

Curved (rotated) or labels aligned to lines cannot be printed n Windows 9x systems if the Print using GDI+ option is turned off in the Tools - Options - Printing dialog.

Enterprise Edition

Enterprise Edition enables Display Options for read-only shared components, allowing us to view Display Options but not to modify them.
Highway Shield Labels

Manifold provides "shield" label styles that may be used to label highways and other roads. The highway shield label styles are scalable: increasing the size of the font will automatically increase the size of the label.

Highway shield labels, like other graphical label styles (such as box styles) can only be used with point labels. To use these to label lines such as highways or other roads we first create point objects from the highway lines and then we create labels from the point objects. It is easy to create point objects by using a convenient transform such as the Centroids or Points transform operators. For example, we could select the highway lines to be labeled, use the Centroids transform on the selection to create points (which will be selected) at the centroid of each highway line segment. We can then cut the selected points and paste them as a new drawing of points and then create a labels component from those points, which can then be formatted using a highway shield label style.

Labels in Manifold System normally paint the text contained within the label in foreground color. The highway shield label styles include four label styles that also use foreground color to paint the label text, but they also include four more styles that paint the label text either in black or in white. This allows the creation of three-color labels, which use foreground or background color and which also allow choice of either black or white as the text color through use of the appropriate label style.

Let's consider a sequence of examples using shield label styles, taken in the order in which they occur in the label choice pull-down dialog. Note that when label styles are reduced to fit into the format toolbar there will be only slight differences between the visual appearance of the different styles. The examples below show the format toolbar on the left and the resultant appearance of the label on the right. Styles are shown using red for foreground color and blue for background color. The first three styles are also shown using other colors to provide examples of different effects that may be obtained.

This first style uses foreground color to paint the text as well as the outline. It works well using black foreground and white background as a label for state highways.

This style also uses foreground color to paint the text as well as the outline and it also adds an upper portion of foreground color. It works well using black foreground and white background as a label for state highways.

This style uses foreground color to paint the upper part of the shield and background color to paint the lower part of the shield. Text is always painted in black. When the same color, such as green, is used for both foreground and background the shield appears in solid color, a good style for regional or state highways.
This style uses foreground color to paint the upper part of the shield and background color to paint the lower part of the shield. Text is always painted in white. When red and blue are used the result is very similar to the symbol used for US interstate highways.

Another style using foreground color to paint the text as well as the outline.

Another style using foreground color to paint the text as well as the outline plus an upper portion of foreground color.

Another style using foreground color to paint the upper part of the shield and background color to paint the lower part of the shield. Text is always painted in black.

The classic interstate highway system label, using foreground color to paint the upper part of the shield and background color to paint the lower part of the shield with text always painted in white.

Notes

Highway shield labels have been designed to work with True Type fonts using regular font settings. Some fonts with atypical proportions or with effects such as boldface may extend outside the label shield.

The initial set of such label styles is designed for use with the US interstate highway system and may be extended in future Manifold releases to include label styles used in other countries.

See Also

Labels
Label Display Options
Repositioning Labels

Only unbound labels may be repositioned. Bound labels are always automatically created at the centroids of their parent objects.

There are two ways of moving unbound labels to a different position:

- Dragging labels by their tie points to reposition them.
- Reposition an entire layer of labels in a map by repositioning the layer.

See the Repositioning Layers topic for information on repositioning the layer. This will move all the labels in the layer together.

To move a label:

1. CTRL-ALT-click on the label to select it for editing.
2. Drag the label to a new position.

Clicking on a label in general means to click on the tie point for that label. Usually the click radius is such that clicking on a label will correctly be interpreted for the label desired. However, in the case of large labels or many complex labels we might want to make the tie points visible to allow us to click more precisely on exactly the label desired.

In that case, we can show tie points in the labels component window by turning on View - Structure - Tie Points.

CTRL-ALT-click on a label tie point to select that label for editing. We can then drag and drop the label to a new position. To make it easier to click on labels, set Edit - Snap to Labels so the cursor snaps to tie points. We can then drag and drop the label in a new position by using

Example

We've manually populated a map of Mexico with place names using an unbound labels component. We then displayed tie points using View - Structure - Tie Points.

![ZACATECAS](image)

We begin by verifying that the mouse is not busy with a command mode (like zoom in or zoom out). CTRL-ALT-click a label's tie point to select the label for editing.

![ZACATECAS](image)

When we click and hold down on the tie point a small box cursor appears.

![ZACATECAS](image)

As we drag with the mouse the box cursor moves with us to show the location where the label will be dropped.
Releasing the mouse moves the label to the new position.

**Example**

Line labels require a **SHIFT** click and drag to move.

Consider a line label created on the border between two provinces. We can select it for editing by **CTRL-ALT**-clicking it, which makes edit handles appear at the coordinates that define the line label.

To move the line label, we **SHIFT**-click and drag any edit handle.
Pressing **SHIFT** while clicking and dragging an edit handle will move the entire line label.

What happens if we do not press **SHIFT** while clicking and dragging an edit handle? In that case, only that edit handle is moved and the shape of the line label is adjusted accordingly.

We can click and drag point labels by dragging the edit handle at their tie point, because the location of point labels is specified by a single coordinate, that of the tie point. Whether we click and drag that edit handle or we press **SHIFT** while clicking and dragging that edit handle the effect is the same since there is only one edit handle to move.

Line labels, in contrast, are defined by locations with more than one edit handle. If we click and drag any single edit handle only that single edit handle moves. If we want to move all of the edit handles together, we press **SHIFT** while clicking and dragging on any edit handle. In this case pressing the **SHIFT** key is a mnemonic to "shift" the entire label.

**See Also**

**Aligning Labels**

**Editing Objects** - Many of the selection and editing methods that apply to objects in drawings also apply to labels.
Rotating Labels

Labels in Manifold that are created from fields may be rotated using the Label Rotation Angle button in the format toolbar for labels. This topic assumes the reader is familiar with labels in general and with the techniques set forth in the Creating Labels from Fields topic.

To rotate all of the labels in a given labels component, open the labels component, press the Label Rotation Angle button and choose a rotation value. To rotate only some labels, turn on per-label formatting, select the labels to be rotated and then specify the rotation angle for those labels. See the Formatting Individual Labels topic for details on formatting some labels but not others.

Example

Suppose we have a labels component containing labels created from a drawing using a Name field. The labels show the locations of towns.

Redmond Oshkosh Marblehead
Palo Alto Wichita Round Rock

We would like to rotate all of the labels by 70 degrees.

To do so we first click on the Label Rotation Angle button.

In the pull-down menu, we choose 70. If we desired some value other than those shown in the pull-down menu, we could choose the More button and specify a value from 0 to 360.
After we choose 70 all of the labels will be rotated by 70 degrees.

Note that the orientation of labels is specified from 0 degrees to 360 degrees as seen above. A label with 0 rotation is in the normal, upright position. A label with 180 degree rotation is upside down.

**Thematically Formatted Rotation**

We can also thematically format labels so that the rotation of each label is controlled by a data field. To do so, click on the Label Rotation Angle button.
In the pull-down menu, choose Theme.

This opens the same Format dialog that is used for thematic formatting of many display characteristics in Manifold. See the Thematic Formatting topic for details on this dialog. For this example, we will use a Population field in our map with an Equal Count method to arrange the labels into five intervals. The rightmost column gives the rotation to be assigned to each. We specified the rotation values by double clicking into each value and entering a number like 0, 60, 120 and so on.
When we press OK the labels will be thematically formatted into different rotations based upon the Population field for each label.

**Example**

Suppose we have a drawing, called "Bearing Lines" with three lines, shown here formatted using a directional line style with an arrowhead indicating the direction of the line.

![Drawing](image)

The drawing's table shows that each line has a Name field containing values like "Track A," "Track B" and "Track C." The Bearing (I) intrinsic field is turned on for display. It has been formatted to show only integer values.

<table>
<thead>
<tr>
<th>Bearing Lines Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
</tr>
<tr>
<td>-----</td>
</tr>
<tr>
<td>35067</td>
</tr>
<tr>
<td>35868</td>
</tr>
<tr>
<td>35969</td>
</tr>
</tbody>
</table>

We will create labels for each line. To do so, we right click on the Bearing Lines drawing in the project pane and choose Create - Labels.
In the Create Labels dialog we give the new labels component the name of "Bearing Labels," we double click on the Bearing (I) intrinsic field to add it to the label text and then we press OK.

If we show the labels component together with the lines drawing in a map, we can see the labels created for each line are positioned over each line.

For a cleaner effect, we can use a box style for the labels. We can format the labels in the map by clicking on the labels layer tab to make it the active layer.

In the format toolbar we set background color to white and then we click on the style well and choose a box style for the label.
The result is a nice box label style that makes it easy to read the label. Now that we would like to rotate each label according to the bearing of each line that it labels.

To do so we click on the Label Rotation Angle button and choose Theme.

In the Format dialog, we choose Bearing ($) as the controlling field and use the Unique Values method. For each value we click into the label rotation box and set the rotation to be the same as the bearing. We press OK.
Each label will then be rotated by the number of degrees in the bearing.

If we like, we can use a different label text. For example, we could open the Bearing Labels component and then choose Labels - Text to edit the text used for the label.

In the Bound label Text dialog we could delete the [Bearing (i)] specification that generated the bearing text and double click on Name to use this field to create the label for each line, labeling it with "Track A," "Track B" or "Track C."

Techniques like the above can generate a very wide range of label styles and effects.

Using different label styles, for example, we can easily alter the appearance of the labels.
A Short Cut

In the illustration showing the Format dialog we had only three lines and so it was easy to manually enter the values 116, 166 and 54 as rotation values to correspond to the bearings of the lines. If we had hundreds of lines we would like this to be done automatically.

If desired, we can also rotate each point based upon the value of the Bearing (I) field using thematic formatting. To do so, press the Label Rotation Angle button and then choose Theme. In the thematic formatting dialog that follows, choose a thematic format that sets the rotation angle by the Bearing (I) field. The fastest way to do this is to use the Continuous shading option in the thematic format dialog.

Using the values of the Bearing (I) field to control rotation:

1. Open the drawing, click the Label Rotation Angle button and choose Theme.
2. In the Format dialog, choose Bearing (I) as the field and any interval-based method, such as Equal Count as the method.
3. Choose 2 as the number of breaks.
4. Check the Continuous shading box.
5. Press the Tally button to establish two intervals.
6. Double click into the two Values cells and change the upper one to 0 and the lower one to 360.
7. Double click into the rotation values and change the upper one to 0 and the lower one to 360. Press OK.

The above procedure creates two breaks, of 0 and 360 degrees and then continuously formats all values in between. In effect, this assigns a continuous range of rotations from 0 to 360 degrees to any Directions values between 0 and 360.

**Example: Adjusting Rotation with Active Columns**

The illustrations above and below show how the text line of the label is perpendicular to the bearing of the line, because a zero rotation shows text in its normal reading position.

If desired, we can use Active Columns to create a derived Bearing field that can be used to rotate the labels at any angle relative to the lines.

![Add Active Column dialog box](image)

We begin by opening the drawing's table and adding an active column. We'll call the column as well as the function that drives it Bearing.
For a script we will provide a simple routine that subtracts 90 degrees from Bearing (I) while "wrapping" at the 0 mark so that resulting values are always between 0 and 360.

```vbnet
Function Bearing
    b = Record.Data("Bearing (I)")
    b = 90 + b
    If b < 0 Then
        b = b + 360
    End If
    If b >= 360 Then
        b = b - 360
    End If
    Bearing = b
End Function
```

For efficiency, we used the on user request setting of Compute, so to see the values computed by the active column we right click on the Bearing column heading and choose Recompute. We can see that the values of the Bearing active column are indeed 90 degrees counter clockwise from the values in the Bearing (I) column.
To use these values to control rotation, we open the thematic format for the label rotation and use the **Bearing** field instead of the **Bearing (I)** field.

The result is that labels are displayed with rotation angles that match the bearing of the lines. Lines with a bearing of 270 degrees will show the text right side up.

However, it could be that we prefer to have labels match the direction of the line but appear as much right side up as possible.
This involves a more complex routine that has a different arrangement for each quadrant between 0 and 360 as implemented above. If we use the above script for the Bearing function we get a different visual effect.

\[
\begin{align*}
\text{Function Bearing} \\
& \quad b = \text{Record.Data("Bearing \ I")}
\end{align*}
\]
\[
\begin{align*}
\text{If } b & < 90 \text{ Then} \\
& \quad b = b + 270 \\
& \quad \text{Bearing} = b \\
& \quad \text{Exit Function}
\end{align*}
\]
\[
\begin{align*}
\text{End If} \\
\text{If } b \leq 180 \text{ and } b \geq 90 \text{ Then} \\
& \quad b = b - 90 \\
& \quad \text{Bearing} = b \\
& \quad \text{Exit Function}
\end{align*}
\]
\[
\begin{align*}
\text{End If} \\
\text{If } b \leq 270 \text{ and } b > 180 \text{ Then} \\
& \quad b = b + 90 \\
& \quad \text{Bearing} = b \\
& \quad \text{Exit Function}
\end{align*}
\]
\[
\begin{align*}
\text{End If} \\
\text{If } b < 360 \text{ and } b > 270 \text{ Then} \\
& \quad b = b - 270 \\
\text{End If}
\end{align*}
\]
\[
\text{Bearing} = b
\]
\[
\text{End Function}
\]

The result is that labels appear mostly right side up, parallel to the lines, regardless of which way the lines point.

Note that the above discussion is helpful for understanding how rotations and active columns work, but the last effect can be achieved automatically within Manifold by using the Align labels to lines option when creating the labels in the first place. There is no need to use an active column or write a script to show labels mostly right side up when aligned to lines.

See Also

Thematic Formatting
Thematic Formatting Example
Thematic Formatting and Labels
Rotating Point Styles
Labels

Editing Labels

Labels in Manifold have much in common with objects in drawings: they are formatted using the format toolbar, they are selected using the same methods that are used in drawings and they are edited using techniques analogous to those used to edit objects in drawings.

Label text may be change. The positions of point labels may be changed, and both the position and the shape of line labels may be changed, even if the label is bound to a drawing object.

Selecting Labels

Labels may be selected within a labels component window or within a labels layer in a map just like objects are selected in a drawing. All standard selection commands will work with labels. See the Selection in Drawings topic for examples of selection methods in Manifold. In particular, Smart Mouse Selection is often used.

Once selected, labels may be Cut, Copied and Pasted in the usual way. For example, a group of labels may be copied from one labels component and pasted into a different labels component. Bound labels may cut if they are not synchronized and they may be pasted into a new unsynchronized labels component bound to the same drawing.

Types of Labels

Labels are either point labels (the default) or line labels.

- Point labels are defined by the coordinates of an imaginary point that provides the center location to which the label's text is aligned.
- Line labels have their position and orientation defined by the coordinates of an imaginary guideline.

The text of a line label is arranged in the shape of the line that defines its position. Point labels are always aligned North up. Line labels always follow the orientation of their guidelines. Neither the points that define point labels nor the guidelines that define line labels are visible except when the label is chosen as the primary selected object for editing, or unless the labels component window is opened and the View - Structure command is used to show either Tie Points or Tie Lines.

Editing Label Text

How we edit text in labels depends on how the label was created.

- To edit the text of an unbound label we right click on it and choose Edit in the context menu.
- If a label is a bound labels component created from fields in a drawing, we change the text by opening the labels component and choosing Labels - Text.

Editing Labels

To change the position of a label we first select it for editing. We do this using the same methods used for drawings as described in the Editing Objects topic: using smart mouse selection (that is, with the mouse not occupied in any command mode) CTRL-ALT click on the label to select it for editing.

The easiest way to select labels is to first use View - Structure to make their tie points visible. One can then quickly click on the tie point to select the label. The illustrations below omit the tie point for greater clarity.

CTRL-ALT click on the label to select it for editing.
An edit handle will appear at the center of the label if it is a point label. We can move the label by clicking and dragging on the edit handle. If the label was a line label, there would be multiple edit handles at the coordinates of the guideline the label follows. We could reshape the line by clicking and dragging the edit handles, or move the entire label with a **SHIFT**-click and drag of any of the edit handles.

If we like, we can add or delete coordinates (and thus, edit handles) to exactly control the shape of the line. To add a coordinate we right click on the line segment where we would like it to appear and then choose **Coordinates - Add** from the context menu. A new edit handle will appear in that line segment. We can then click and drag that edit handle to move it as desired.

### Context Menu Editing Commands

Right clicking on a label that has been selected for editing calls a context menu that contains editing commands. When right clicking on an editing handle on a label selected for editing, or onto a line segment between editing handles, the context menu will also include additional editing commands.

#### General  
(Right clicking onto label)

- **Duplicate**
  Duplicate the label. It is wise to immediately move the new label while it is selected. This avoids creating coincident labels that might later cause confusion.

- **Center**
  Pan the view to center this label in the window.

- **Coordinates**
  Launches the Object Coordinates dialog to allow direct editing of the coordinates that comprise the label.

#### Branch  
(Right clicking onto a label selected for editing)

- **Delete**
  Delete this branch. Not enabled unless there is more than one branch.

- **Duplicate**
  Duplicate this branch at exactly the same position. Use **CTRL-SHIFT** to drag to a different position the new branch that is thus created (which otherwise exactly overlies the previous branch).

- **Split**
  Split this branch into two branches at the clicked position.

#### Segment  
(Right clicking onto a label selected for editing)

- **Delete**
  Delete the line segment clicked along with the edit handle coordinates to either side of the segment. Not enabled if there is only one segment left.

- **Delete / Split Branch**
  Delete the line segment clicked along with the edit handles to the left and right and also split this branch into two branches at that location.

#### Coordinate  
(Right clicking onto an edit handle)

- **Delete**
  Delete this coordinate. It is not possible to delete the last two coordinates defining a line label.

- **Delete / Split Branch**
  Delete this coordinate and also split this branch into two branches at that location.

- **Duplicate**
  Duplicate this coordinate. It is wise to immediately move the new coordinate/edit handle to avoid confusion that might be caused by coincident (redundant) coordinates.
Labels

**Coordinate**  (Right clicking onto a segment between edit handles)

- **Add**  Add a new edit handle (coordinate) at the location clicked.
- **Add on Segment**  Add a new edit handle (coordinate) on the existing line near the location clicked.
- **Add Mid-Segment**  Add a new edit handle (coordinate) at the middle of the segment nearest to the location clicked.

**Examples**

The following sequence of illustrations use line labels created for a drawing that has three lines, which illustrate three trails in a park. Initially, we see the labels in a labels layer in a map above the drawing.

![Diagram of labels in a map](image1)

We begin by **CTRL-ALT** clicking one of the labels to select it for editing.

![Diagram with edit handles](image2)

Edit handles appear.
To move the entire label, we **SHIFT** click and drag any of the edit handles.

This moves the label to a new position.

If we deselect the label we can see the text has moved.
Let's now shift gears and open the same labels component in its own labels window. When the labels component opens in its own window the objects from its parent drawing (this is a bound labels component) will appear in gray schematic color. The appearance is similar to the view in the map because the drawing is a very simple one.

We've opened the component in its own labels window because only in a labels component window can we use View - Structure - Tie Lines to see the guidelines that specify the geometric shape of the label.

We can see from the above illustration that when a bound line label is created the guideline for it is the same as the parent line object in the drawing to which the label is bound. However, even though that is how the label is created, once the label is created if desired we can edit the metric, which is the geometric definition of the label shown by the guideline, independently of the shape of the parent line object.

For example, we can delete some of the coordinates defining the guideline.

To do so we CTRL-ALT click on the label to select it for editing and then we right click on the coordinate / edit handle we would like to delete.
In the context menu that pops up we choose **Coordinate - Delete** and the coordinate will be deleted.

The illustration above shows the result after we have deleted several coordinates at the beginning and end of the guideline.

If we deselect the label so the guideline is more visible, we can see it is now smaller than it was originally.

Of course, if desired, we can delete coordinates in the middle of a label's metric as well.
Consider the above example. We’ve selected the label for editing and then we have clicked and dragged three of the middle edit handles to reshape the guideline and thus the shape of the label. We would now like to delete one of those coordinates and so we right click on it and choose Coordinate - Delete in the context menu that pops up.

If we would like to delete many coordinates at once we can use a quick trick, of splitting the label into two branches and then deleting one of the branches.

For example, we can right click on the edit handle seen above to split the label into two branches at that position.

In the context menu we choose Branch - Split to split the label into two branches.

The only visible evidence that the label is now split into two branches is quite subtle: as seen in the illustration above the label text appears to have been pushed into the left branch and there appears to be no text in the right branch. What is going on is that normally both branches would have the label text repeated in them; however, the branch on the right is too short to host the text at the given zoom level (the text won’t fit at this font size and zoom level) and so the text is not displayed on the branch on the right.

We can delete the branch on the right by right clicking it.

We choose Branch - Delete in the context menu and...
...that deletes the branch on the right.

Splitting a line label into two branches to delete one of them as a short-cut way of rapidly editing a label is usually the limit of sophistication for most people in working with branches. The idea of the same object appearing to be in more than one place at the same time (which is what can be done with branches) can very rapidly become confusing to the point of uselessness. Nonetheless, just as objects in drawings can exist as multi-branched objects labels can also so exist.

Let’s ease into this part of the example by considering the labels component above, shown in its own window. We will **CTRL-ALT** click a label to select it for editing.

Now, we will **SHIFT** click an drag the label to move the entire label to a new position.

We now **right click** on the label to launch the context menu.
In the context menu, we choose **Branch - Duplicate** to duplicate the single branch of the default label into two branches. The second branch is a copy of the first and exists at exactly the same position.

Since it is weird to have two branches of a label at exactly the same position and thus easy to forget this label has two branches located at exactly the same place, the first thing we do is move one of the branches by using **CTRL-SHIFT** click and drag.

This shows what now appear to be two labels, but both are simply different branches of the same label.

If we deselect the label we can see the same text is repeated on both branches. This is the result of the default **View - Display Options** setting to **Label each branch**, which is checked by default.
Even though broth branches are part of the same labels object, we can edit each branch independently of the other. For example, in the illustration above we have CTRL-ALT clicked the label and then moved the edit handles for the lower branch by clicking and dragging them to change the shape of the lower branch.

This results in a different shape for that branch. It is rare that we need to use multiple branches for labels, but if need be this technique may be used. The point of the example above is that for editing purposes line labels behave very much line lines, with the same editing techniques being applicable.

**Snapping Labels to Grids or Graticules**

Snap to Grid and Snap to Graticule modes may be used when repositioning or otherwise editing labels. Usually these modes are employed with use of the spacebar to toggle snap mode off and on while editing.

See the Edit - Snap To topic for an example using Snap to Grid to reposition labels.

**Non-GDI+ Printing Limitation**

Curved (rotated) or labels aligned to lines cannot be printed in Windows 9x systems if the Print using GDI+ option is turned off in the Tools - Options - Printing dialog.

**Enterprise Edition**

There are several nuances involved in working with labels components in Enterprise Edition arising from the normally read-only nature of labels that are shared components. Commands that don't make sense for read-only components (such as cutting and pasting) are disabled. Note that if a labels component is a bound labels component it is generated based upon a drawing and its table. Any commands that remove label objects from a bound labels component (such as Cut, Delete and Delete All) are disabled in Enterprise Edition.

Enterprise Edition enables Display Options for read-only shared components, allowing us to view Display Options but not to modify them.

**Converting Labels to Points or Lines**
Suppose we have a labels component consisting of unbound labels. These may have been created by hand or imported from some format as an unbound labels component. If we need to, we can create a drawing from that labels component by in the project pane Copying the labels component and then right-clicking into the project pane and choosing Paste As - Drawing. This will create a drawing with point and line objects for point and lines labels with a Text attribute in the drawing's table for each object giving the label text.
Legends
Legends

Before reading this topic, please read the Adding Legends topic for an introduction to legends.

The Legend Dialog

The Legend dialog allows us to create and to customize legends. Open the dialog by choosing View - Legend at any time the component is open for which the legend is desired. If the legend is already displayed in addition to using View - Legend we can call the Legend dialog by double clicking the legend or by right clicking the legend and choosing Properties from the context menu.

To show a legend we simply check the Show legend box in the Legend dialog to show a default legend created by Manifold. We may customize the default legend to add, delete or modify items shown in the legend. To customize a legend, we start with the default, automatically created legend. In the Legend dialog, we check the Customize legend box and then edit the legend as desired using the toolbar controls as well as options in the Customize legend section of the dialog.

Legends are edited by adding or deleting elements in the Legend dialog and then highlighting those elements and specifying their options.

Legend Dialog Guided Tour

Let's take a tour of the different parts of the Legend dialog.

Consider the legend above, a typical legend as might appear for a drawing that uses thematic formatting to color areas by a Population field.
If we double click the legend we see the **Legend** dialog above. Let's examine the different parts of this dialog.

The upper left region of the dialog contains controls that set the overall appearance of the legend.
Looking more closely we see this part of the dialog allows us to choose the **Style** of legend as well as allowing us to provide any **Caption** if desired. The colors specified will be used as foreground and background colors throughout the legend style specified.

The right hand side of the dialog consists of a pane in which legend elements are listed together with a toolbar of command buttons that operate on legend elements.

Taking a closer look at the elements pane we can see that it has an element for each item that appears in the legend. Highlighting an element by clicking on it will select it as the subject for any commands or option choices. In the above illustration we have clicked on the first element, a text element reading **Mexico Drawing**, to highlight it. Commands such as the toolbar **Delete** button will now apply to that element.
A right bracket ] shows which lines in the pane are all part of the same element. Some elements consist of only a single line. For example, the first element consists of a single line. The second element is a dynamic element that consists of more than one line. It is an automatically-generated element that reports the thematic formatting for areas. As a convenience for the user, Manifold will automatically create such elements if desired so that we don’t have to manually enter many lines of legend elements and manually specify characteristics such as colors and interval numbers for each such line.

The third and fourth elements are also dynamic elements that could show formatting, but because there are no points or lines in this drawing they show nothing. The fifth element is another text element consisting of a single line of text, the copyright notice.

When we choose an element by clicking on it to highlight it, the lower left portion of the **Legends** dialog will display options for that element.

For example, clicking on the first element to highlight it...

...will display the options related to a text element. In this case, the system shows the **Type** of the element and the **Text** that is to appear. Pressing the font control button to the right of the **Text** box will open a subsidiary dialog allowing us to specify the font characteristics (such as font, font size, attributes such as bold facing, etc.) of the text.
To see the settings for a different element we can click on it to highlight it. For example, we might click on the second element. Because this is a dynamic element all of the lines involved in the element will be highlighted when we click on this element.

The settings and options for this element are considerably more extensive than for a text element. This element type, *Format: Areas*, is one that automatically reports the formatting of areas. Options allow us to hide the element if there are no areas in the drawing and to specify what formatting characteristics should be shown in the area samples. For example, checking the *Background* box (it is checked by default) will cause this element to display thematic formatting for *background color* for areas.

**Legend Dialog Controls**

- **Show legend**: Check to show a legend in the active window.
- **Align**: Location of the legend within the display window. Choose *None* to allow repositioning of the legend by dragging.
- **Style**: Overall appearance of the legend. Use the *Transparent* style for a legend without borders and transparent background color.
- **Align caption**: Align caption to the right side, center or left side of the legend.
- **Align text**: Align text within elements to the right side, center or left side of the legend.
- **Caption**: Legend title. If no text is provided for the caption, the title bar will not appear. The font button to the right of the Caption box allows font customization for the caption. The default font is taken from the *Legend Caption* setting in the Tools - Options - Fonts pane.
- **Margin**: Blank space in points to be inserted between the body of the legend and the surrounding border or any separators. 1 point = 1/72 inch.
- **Colors**: Colors used for the legend border and body fill.
- **Customize legend**: Check to edit the elements shown in the legend preview pane.

- **Add Element**: Create a new element in the legend. A new element is always created as a text element. This can then be changed to the desired type of element in the Type box.
- **Add Samples**: Enabled for images. Create the desired number of color samples for the most frequently used
colors in the image.

**Delete Element** - Delete the highlighted element.

**Move to Top** - Move the highlighted element to the top of the legend.

**Move Up** - Move the highlighted element up one position in the legend.

**Move Down** - Move the highlighted element down one position in the legend.

**Move to Bottom** - Move the highlighted element to the bottom of the legend.

**Flatten** - Convert a composite legend item into the equivalent set of static legend items. For example, a set of automatically-created color boxes for a thematically formatted height in a surface could be converted into individual static elements.

**Type**
Enabled when customizing the legend. Choose the type of object sample to display in the legend from a variety of dynamic or static types of elements. Also available are vertical or horizontal separator lines, labels and text only lines. Use a blank text line to provide a blank vertical space between legend elements. Inserting a horizontal separator line will break the legend into multiple columns.

**Text**
Edit the main text string displayed as part of the element. The legend box will expand to fit the text provided.

Font selection for label strings and text strings used in elements.

### Adding an Element to a Legend

To add a new element to a legend we press the **Add Element** button and a new text element will be created just above whatever was the currently highlighted element. We then change the type of the element to that desired and move the element to the desired position.

#### To add an element to a legend:

1. Open the **Legend** dialog for the legend.
2. Check the **Show legend** and the **Customize legend** boxes.
3. Press the **Add Element** button in the **Legend** dialog's toolbar.
4. A new text element will be created just above the previously highlighted element. The new text element will be highlighted.
5. In the dialog's **Type** box, choose the type of element desired and choose any desired options from controls that become available for that type of element.
6. Move the new element to the desired position in the stack of legend elements and press **OK**.

To delete an element, simply highlight it and press the **Delete** key.

### Types of Legend Elements

There are two main classes of elements used in Manifold legends:

- **Dynamic elements** are automatically generated by the system and are used to show thematic formatting. Such elements are usually composite elements that combine multiple legend lines consisting of text and samples. Dynamic elements normally include samples that show thematic
Legends

formatting as well as a text line providing an overall label. Because dynamic elements take their content from the parent component automatically, their formatting (such as colors used) cannot be changed manually.

- **Static elements** are manually created and controlled. All parts of static elements may be customized, including the appearance of samples such as sample style and color. Static elements are items such as text lines, separators and manually specified sample elements. For faster workflow, we might often create static elements by first letting the system create a dynamic element and then flattening it into static elements, which we can then edit as desired to fine-tune appearance. This is often faster than manually creating static elements one-by-one.

Sample elements consist of a graphical sample that shows a color or other formatting sample and a text that explains what that sample means. For example, an element illustrating a particular thematic format for an area may include a sample of area color in the form of a small rectangle together with a text that identifies what that color means. Sample lines are normally part of dynamic elements. A wide variety of sample types is available for manual creation of static sample elements.

Let's look at examples of the above.

Consider a legend similar to that used for our tour. We have added points to the drawing and have also added a legend element for those points.

Opening the **Legend** dialog we can see that the **Copyright 2006** element is a typical **text** element consisting of a single text item with no sample. This is a static element because it is created manually.
The element for points is a typical sample element with two items, a sample and a text. The sample in this case is a sample graphic of what a point looks like in the drawing given the formatting currently applied. The text is the default text used by Manifold whenever a point format type is added, Points.

To see if the element showing points is a static element or dynamic one we can look at the options for this element. It shows that the Format: Points type is used for this element, so it is a dynamic element that now has a very simple, one-line appearance because there is no thematic formatting used for points in this example.

Legend element types that begin with a Format: are always dynamic elements because they are called upon to show thematic formatting.

If the element for points was a static element it would have used the type Point: Single type and instead of a list of formatting characteristics to show the options would consist of formatting wells allowing us to specify the foreground color, background color, style, size and rotation angle of the sample that appears in the element.

The dynamic element showing area formatting is a typical dynamic element. It consists in this case of several sample lines showing area formatting and a text line, Population. Note that even though this element consists of several lines it is still only a single element which when clicked becomes highlighted in its entirety.
If we highlight that dynamic element and press the Flatten button in the legend dialog's toolbar, we can convert the dynamic element into a series of individual static elements.

In this case, the individual elements are no longer automatically created by the system but have become static, manually controlled elements.

Flattening is a one-way process. Once a dynamic element has been flattened into individual, static elements, we cannot then "un-flatten" those individual static elements back into an automatically-created dynamic element.

Samples

Samples are small graphical elements, such as a swatch of color or a point icon, that show colors and other formatting characteristics in a particular legend element.

In drawings, samples are used in elements that describe areas, lines, points and labels. In images samples are used to show heights. In images, samples are used to show sample color swatches of colors that appear in the image. These elements have a graphical element on one side of the legend and text that describes what that sample means on the other side of the legend.

When static elements using samples are highlighted the dialog will provide controls in the lower left corner of the dialog that allow us to specify the appearance of the graphical sample used as well as the content and appearance of the text. Controls will be different for different types of samples. For example, point samples will have a rotation control that allows us to rotate the sample icon whereas sample elements for areas and lines do not have a rotation control.

Dynamic elements that show samples do so automatically with the appearance of the sample being automatically taken from the formatting in use. Dynamic elements that show objects with thematic formatting will show a range of samples to illustrate the thematic formatting and the intervals involved. Dynamic elements that show object formatting that does not use thematic formatting will illustrate the formatting in a single line element like that of a static element.

Quite often the fastest and easiest way to create a manual sample element is to first create a dynamic element (or use the default dynamic element created for a component by Manifold) so that formatting for samples is done for us automatically by the system. We can then Flatten the dynamic element into one or more manual sample elements that have the samples pre-loaded with formatting options. We can then adjust those samples as we like without having had to take the time to create them in the first place.

Dynamic elements in Drawings

Dynamic elements are used to show formatting in drawings. They will automatically create as many lines as are necessary in the legend to show all formatting attributes. Additional lines will be added as necessary if additional thematic formatting is added to the drawing.

Example

This example shows different features of a dynamic element. We begin with an empty legend and add a dynamic element showing area formatting. We show the effect of different options on that element.
We then flatten the dynamic element into numerous individual, static elements and edit those elements to achieve the legend appearance we desire.

Let's begin with the sample drawing of Mexico.

The drawing has been thematically formatted using the **Population** field to color background color for areas as seen above. There are no lines or points in this drawing.

We open the **Legend** dialog for this drawing. For the sake of this example, we have checked the **Customize legend** box and have then deleted the default elements appearing in the legend. That allows us to start with a blank legend to show the creation of a legend from the very beginning.
To add a new element to the legend we click the **Add Element** button.

This adds a new element, a **Text** element. New elements are always created as **Text** elements. If we desire a different element we highlight the **Text** element and change its **Type** to whatever type of element we want.
For example, in the Type box we can choose Format: Areas as the new type for this element.
The element will immediately be converted to a **Format: Areas** element, which by default shows a report of the formatting used for areas. The element consists of a small text line (which can be eliminated by deleting the text in the **Text** box for this element) together with samples showing thematic formatting in use.

The lower left corner of the dialog shows options used for configuring this type of legend element.

The lower pane lists all formatting characteristics that apply to areas and which can be shown in a legend of this type. The **Background** check box, for example, when checked will cause the element to show formatting for area **background color**. The other boxes list in alphabetical order all format options for areas.
If we scroll down in the option box we can see that every formatting characteristic available for areas is checked by default. Although all area formatting characteristics are listed the legend currently shows only the thematic formatting for area background color because that is the only thematic formatting that has been applied.

If we click the **OK** button a legend will be created in the drawing as seen above. The legend shows the result of the single element in this legend, the **Format: Areas** element. Because only area background color has been thematically formatted the legend is a relatively simple one that shows the range of colors and values given in that thematic format.

If we add thematic formatting for other formatting characteristics in this drawing, the legend will show them as well. For example, we can click into the area style formatting well in the format toolbar and specify a thematic format for area style.
In the above example we have used the TOTVEH_199 field, which gives the total number of vehicles in each province, to format style using a variety of styles that strike our fancy. Note that the Continuous shading box has been un-checked since we don't want interpolation between styles, which would result in styles being used that are in between those shown above.

Well, that's an interesting effect! Each area in the drawing is now formatted so that the color is set by a thematic format depending upon the total population in the province while the style used for the area is controlled by thematic formatting based upon the total vehicles in the province.

Looking at the drawing's legend, we notice one more interesting thing:
The legend in the drawing has automatically expanded to include a second set of samples that show how area style varies in the given intervals. This happened automatically when we specified the additional thematic format for style without our needing to do anything in the Legend dialog.

To see what is going on, we can double click the legend to open the Legend dialog.

The first thing we notice is that the single element in the legend, the Format: Areas element, has expanded to include multiple additional lines. It has expanded to report the thematic formatting of area style as well as area background color.
Note that the interval numbers shown for area background formatting are single numbers because that thematic format used the **Continuous shading** option. In contrast, the numbers reported for the style formatting show a range of numbers for each interval because the **Continuous shading** option was not used in that thematic format.

When we added thematic formatting for style, this legend element expanded to include **Style** because that is one of the boxes checked in the options pane for this **Format: Areas** element. We can un-check that box to see what happens.

If we scroll down to the **Style** box and un-check it, the lines in the legend element that report style formatting go away.

If we press **OK** we see that the legend in the drawing also no longer shows any lines reporting **Style** formatting.
If we launch the **Legend** dialog and check the **Style** box again...

...then once again the legend will show entries for **Style** formatting.

What is going on here is that a **Format: Areas** element is a command to create a legend element that automatically shows lines for all formatting characteristics checked in the option box if they are used for thematic formatting. Any new thematic formatting characteristic that is specified will automatically appear in the legend.
We can see this in action by specifying yet a third thematic format. We will click on the area border foreground color well in the format toolbar and specify a thematic format to automatically change the color of the area borders by changing area border foreground color.

We will use the `BUSES_1991` field that gives the number of buses in each province in the year 1991.

Adding this thematic format dials up the garish appearance of the drawing by yet another confusing notch.
And once again, automatically, a new set of samples appears in the legend to show how area border foreground color varies by number.

Note that when Manifold constructs samples if multiple formats are in play the system tries to create reasonably sensible samples that incorporate the various formatting characteristics specified. In general, each range of samples takes as the non-varying characteristic whatever is at the top of the stack from characteristics being varied.

For example, in the samples of styles since the uppermost style is one without a pattern the samples used to show area background color and border foreground color use no pattern for their style. The style and border foreground color samples use the first color, the darkest, from the area background color range. The samples for area background color and style use the uppermost border color, a hot pink, as their border colors for the samples.

If we double-click into the legend to launch the Legend dialog we can see how it has changed to accommodate this third thematic format we have added.
As expected, it now shows three ranges of samples within the **Format: Areas** element.

If we like, we can make this element shorter by un-checking some of the options.
For example, if we un-check the **Background** and the **Style** boxes the element will show only the range of thematic formats for the area border foreground color.

The legend displayed in the drawing likewise will show only samples for the area border foreground color.

Let's continue this example with an example of using the **Flatten** command, followed by examples showing editing of individual elements to alter the appearance of the legend.
We will double-click the legend to launch the **Legend** dialog. In the dialog we will check all the option boxes for the **Format: Areas** element.

With the element highlighted we press the **Flatten** button to convert the dynamic element into a series of individual elements.
The result is a series of individual elements. Note that even though the **Format: Areas** element was only partially shown the result of flattening includes individual elements for all items that were in the **Format: Areas** element, even those that were not shown because of the truncation.
We can now edit the individual elements. For example, we can select the text lines and change their text into strings like Population, Total Vehicles and Buses for more descriptive titles.
We may decide that a wide range of thematic samples leads to a confusingly large legend. Perhaps we think that only one sample for each thematic variation will suffice. In that case, we can highlight in turn each element we do not want and then press the **Delete Element** button to get rid of it.

Note, by the way, that after flattening each element is now a *static* element, that is, it is manually specified. When an element is highlighted, the options for it become a series of formatting wells that allow us to specify the appearance of the sample.
We can change the appearance of the sample by clicking into the sample wells and changing them. For example, we can change the area background color for the **Buses** sample to default gray.
Likewise, we can click on the sample above it and change the area background color to default gray and the border foreground color to default dark gray.
We click on the topmost sample element and change the border foreground color to default gray as well. The idea is to create a legend that varies only one formatting characteristic at a time from the default gray to emphasize which characteristic is being varied for that parameter.

This results in a simplified legend that shows three samples, one for each formatting characteristic that is varied by thematic formatting. While it is probably a fool's errand to try to explain clearly such a chaotic drawing (too much formatting of too many characteristics at once is difficult to comprehend), there is a possibility that a simplified legend may help people guess what is going on.

Suppose we decide that the numbers to the right of the samples are confusing. It is easy to remove them as well.
To do so, in the Legend dialog we highlight each of the sample elements in turn and then delete the text in the Text box.

This creates a legend with no text to the right of each sample. However, it could be we decide we don’t like the text titles on lines above the sample. We would like to see them on the same line as the sample. That too is easy.
In the **Legend** dialog we delete the text elements between the samples, and then we highlight each sample element in turn and in the **Text** box enter the text we want next to the sample. For example, we have entered **Buses** (using the font button to the right of the **Text** box to boldface the text) into the **Text** box for the lowest sample. Note also that we have chosen **Left** as the option in the **Align text** box to bring the text titles closer to the sample boxes.

That's a step in the right direction, but suppose we would prefer greater spacing between the sample boxes and the accompanying text.
To arrange that we go back to the **Legend** dialog and in the **Text** box for each element we add some leading space characters to the left of each text. This will move the text to the right a bit.

The result is a small but elegant legend that shows how **Population** is indicated by varying area background color, **Total Vehicles** by varying style and **Buses** by varying border foreground color.

Beginners will follow a workflow like that above, going back and forth between the **Legend** dialog and the component window, to create legends via a series of commands that change things until the legend looks right. More experienced users will tend to go straight to the desired effect by manipulating many different settings within the **Legend** dialog at once.

This is a somewhat contrived example because an experienced designer would not create such an inelegant and difficult-to-comprehend mixture of simultaneously-varying thematic formats. It is usually wise to vary only one formatting characteristic at a time for maximum comprehension.

**The Collapse Similar Formats Box**

When the **Collapse similar formats** box is un-checked, Manifold will provide a legend line for each formatting characteristic checked in an option box for elements like the **Format: Areas** type even if that formatting characteristic is not the subject of a thematic format. In the case of a default drawing with no thematic formatting...
applied the result is eight sample lines (one for each of the eight formatting characteristics of an area) that appear as identical samples.

If a single thematic format is in play, such as thematic formatting for area background color, un-checking this box will result in one thematic format range of legend samples plus seven other sample lines that all look alike. Checking the box hides the redundant seven lines.

Why provide a check box at all? In the case of a single drawing this is done so that the box may be un-checked and then the resulting element may be flattened so that each of the additional sample lines may then be manually manipulated. It's just a simple way of producing some extra sample lines, one for each formatting characteristic that may be varied.

**Types of Elements**

In addition to the **Format: Areas** type (and analogous **Format: Lines** and **Format: Points** types) Manifold provides a variety of static types with different styling of samples to show areas, points, lines and labels.

The simplest way to show examples of various type is to show them used in a legend, as above. Each of the above types may be configured manually with formatting, such as choice of colors, to provide a sample with the desired appearance. The **Colors** choices are used when creating legends of colors for images or surfaces, and may also be used to show area formatting in drawing legends as well.
Not illustrated but obvious how to use are the Label: Single type and Text type, which are simply text-oriented types.

There is a Separator type that provides a horizontal line. There is no separator type without a line, since to add an extra row of white space between lines (as seen in the above legend) we simply use a Text type with empty text.

The Vertical Separator and Vertical Separator with Line types break legends into columns, either with a vertical line between columns or without.

Legends in Surfaces

Legends are created for surfaces using the View - Legend dialog just as with other components. Surfaces use one main type of legend element, the Format: Heights type that appears in the list of types when the dialog is opened for a surface.

Example

Consider the above surface, a portion of the Montara Mountain example surface on the Manifold CD.
We have formatted the surface using the above palette in the View - Display Options dialog. Note that there is a sequence of intervals defined, 250 units apart, from 250.00 to 1750.00.
To add a legend we launch the Legend dialog and click Show legend. Checking the Customize legend box shows us the default arrangement for a legend for a surface. The legend consists of one text element giving the name of the component in boldfaced text and a dynamic element of type Format: Heights that shows formatting for the surface. The lines appear at locations specified by the numeric values for intervals given in the thematic format, in this case, at every 250 units of height.

If we press OK we see the legend consists of a continuous bar of thematic colors marked at the upper and lower ends by the upper and lower numeric values of the interval range used for thematic formatting.
Going back into the **Legend** dialog we can try different options for the **Format: Heights** element. For example, we can switch off **Lines** and turn on **Tick marks**. This adds short lines to the right of the gradient color bar instead of lines running through the bar.
We can click on Intermediate labels to show text labels for formatting intervals between the highest and lowest. The values of labels and their locations are defined by the interval numbers specified in the thematic format for this surface in the Display - Options dialog.
If desired, we can choose Left in the Align text box to move the intermediate labels closer to the color bar. We've also turned off Tick marks and have turned on Lines.

The result is a more compact legend.
Let's get back into the Legend dialog to explore other options.
Un-checking the **Use continuous shade** converts the continuous gradient bar into a discrete collection of legend lines, each with a color sample. If we ever apply the **Flatten** command to a **Format: Heights** element for a surface the result will also be a set of static elements like these single line elements.
Let's check the **Use continuous shade** box again. Suppose now we would like to eliminate the **Heights** label at the top of the Format: Heights element. Because this is an integral part of the Format: Heights element we cannot simply highlight it and delete it as we could with a text element.

Instead, we eliminate that part of the element by removing the text in the Text box for the element.
Entirely removing the text from the Text box will eliminate that line from the legend.
Let's close this example by adding a Caption, the text Altitude in 9 point font and then pressing OK to create the legend.

The result is a compact and useful legend that shows the range of heights and their colors.

**Proportional Scale Option**

The proportional scale option for the Format: Heights legend element style for surfaces controls how intervals are represented on the gradient bar. Either the ticks on the gradient bar for each interval are located in proportion
to their numeric distance from each other, or the ticks are arranged evenly at equal distances on the bar and the
color spectrum shown in the gradient bar is adjusted so that colors at each tick appear as they should.

Example

Consider the same Montara example surface, but this time colored differently using a different palette.

Opening the View - Display Options dialog we can see that the surface has been colored using intervals that are
highly uneven. The first interval runs from 250.00 to 1000.00, but the next interval runs only from 1000.00 to
1001.00. The remaining intervals are arrayed to cover a range that is about the same size as the first interval by
itself.
If we open the Legend dialog for this surface we can see that the default arrangement using **Format: Heights** draws a gradient bar with intervals marked at their proportional distance along the bar. That is, the height of the bar for the first interval is genuinely about the same as the rest of the intervals put together. We have added ticks and intermediate labels so the exact placement of intervals is clear.

Note that two intervals are so small there is no room for the 1000 and the 1001 labels as well as the 1035 label, nor is there room for two tick marks for the 1000 and 1001 intervals. Manifold will automatically clip (that is, hide) labels and tick marks for which there is not room.
If we would like a longer gradient bar we can make it taller by increasing the Size to 200 points. This provides a taller bar, which may be more useful if we have many labels we would like to show.
If we un-check the Intermediate labels box the gradient bar immediately shrinks to a standard size and intervals are provided at even distances. Note that the distance between the 1000.00 line and the 1001.00 line is the same as the other distances even though this interval is only 1 unit apart. If we compare this illustration with the preceding illustration, we can see that the color at each interval bar is the same, but that the color gradient has changed to accommodate the even spacing of the interval lines.

**Legends in Images**

By default, legends created for images are empty except for a single text line giving the name of the image.

To add color samples to an image legend, check the Customize Legend box and then use the Add Samples button to add the desired number of color samples. This button is enabled only when working with the Legend dialog for images.

Color samples show color wells for the most frequently used colors. Grayscale and palette images will report an exact result while RGB and RGBA images will report an approximate "most frequent" result based on an analysis of color frequency using a Monte Carlo algorithm. The colors used will be reported in R:G:B format.

Legends with images are useful with land cover and land use data displayed as a color-coded image. In such cases the legend elements can be customized with user-specified text replacing the RGB color captions to provide appropriate text captions for each color well.

**Context Menu for Legends**

Right clicking onto a legend calls up the following context menu commands:
Legends

Hide
Hide the legend. Equivalent to un-checking the Show legend box in the Legend dialog. To show the legend again choose View - Legend in the main menu and check the Show legend box.

Align
Choose the alignment of the legend from Left Top, Left Bottom, Right Top, Right Bottom, or None, corresponding to the Align choices in the Legend dialog.

Copy
Windows clipboard copy operation. Copies the legend onto the clipboard, flattening all legend items prior to copying.

Copy Structure
Windows clipboard copy operation. Copies the legend onto the clipboard, without flattening, so the legend is copied as is. Use Copy Structure when a dynamic legend item, such as a set of formats based on height, is to be preserved.

Paste
Enabled only if a legend has been copied onto the clipboard. Paste the legend in the Windows clipboard into this legend, replacing this legend with whatever is the content or structure of the pasted legend. To paste a legend into a drawing we must first show the existing legend and then right click onto the existing legend and choose Paste.

Paste Append
Enabled only if a legend has been copied onto the clipboard. Like Paste, but adds the items from the pasted legend to the legend that was right-clicked instead of replacing it.

Properties
Launch the Legend dialog for this legend.

It is often the case that we have a group of similar drawings or other components and want to duplicate the look and feel of a finely-crafted legend in many components. The fastest way to accomplish that task is to use Copy or Copy Structure to copy the model legend and to then open each component in turn, show its legend and then Paste into that legend.

Printing

When printing from a layout, a component within a layout can be configured to automatically show the legend it has or a new legend can be created within the layout for that component. See the Layouts topic.

Note that it is often more convenient to create a legend in a layout as a separate layout element, so that it may be easily moved about within the layout, than it is to have the legend appear as part of a map or other component within the layout.

Alignment

The legend may be automatically positioned in the display window or dragged to a given location. Position is set via the Align option in the Legend dialog. If the legend is already displayed we may also right click on the legend and choose the Align context menu for alignment options.

- Left Top
- Left Bottom
- Right Top
- Right Bottom
- None

All alignment options except None will place the legend in the same relative location in the window even if the window is resized or the component is panned or zoomed.
The **None** alignment option fixes the legend at the relative location to which it is dragged by the user. If the window is resized, the legend will move to stay in the same location relative to the size of the window. Note that it is possible to resize the window so that some parts of the legend are clipped by the window's boundary.

**Fonts**

Set the default fonts used by the legend (including the caption) in **Tools - Options - Fonts**. See the Tools - Options topic.

![Use the font control to specify the font for text and labels within the legend.](image)

The maximum size of legend font is **1440** points (approximately 20 inches high). Very large fonts are used when printing legends for very large format layouts. Legends will automatically resize to accommodate large captions. The preview pane will try to show the characteristics of the font but will show text only in the 8 to **12** point characters to preserve the usability of the preview pane.

**Styles**

The **Style** box allows choice between **Standard** and **Transparent** styles. Together with use of the color boxes and the **Caption** this allows us to specify a different appearance for the legend.

![For example, suppose we have a drawing of Mexico that is thematically formatted.](image)

We can add a legend using the **Standard** style.
This creates a legend in a box with an upper caption. The caption box is colored using the same color as the border (black) and the legend box is filled with the fill color (white).

We can change the style to **Transparent** for a different effect.

In this case the border and background disappear and the fill color is used to fill in the caption.
We can remove the Caption text.

In that case the caption disappears (the legend has been moved to a different position in the drawing for this screenshot). This is a nice effect for a simple legend that uses whatever is the background color of the component.

To see how a Standard style legend looks with no text in the Caption box, we change the Style back to Standard.
That will draw a border but no caption. The fill color (white) is used to fill the **Standard** style legend.

If we would like a border but we would like the legend to use the same color as the background in the component, we can change the fill color to a light gray color, the same as is used for the background.

That will display a legend with the same background color and also with a border.

**Interval Styles in Legends**

Thematic formats can be interval based, to assign a formatting characteristic based on a fixed range of interval values, or they can be **continuous** if the **Continuous shading** box is checked. Continuous shading results in a smooth interpolation of color between specified values and colors.
Legends will automatically adapt to the status of the **Continuous shading** box.

If the **Continuous shading** box is checked the legend will show the discrete values and colors between which a continuous range of colors is interpolated.

Dynamic legend elements will use the same number of digits beyond the decimal point that are specified for that column in table column formatting.

**Resetting a Legend**

To go back to the default style of legend, un-check the **Customize Legend** box and then show the default legend. This will clear any customized legend lines.

**Legends in Maps**

By default, legends in maps will automatically be created by combining the legends of the map’s constituent layers, with the legends being assembled in the same order as layers.
Maps have a dynamic legend element type called Component: together with the name of a layer. The Component: element type automatically incorporates into the map whatever is the current legend for that component. The Hide if turned off checkbox hides the Component: element type if that layer is turned off. Legend elements for map layers that are currently turned off are marked hidden in the legend dialog.

Columns

Columns are created by adding a new line to the legend and choosing the Vertical Separator type for that line. See the Adding a Legend example topic.

Adding a Legend to a Print Layout

See the Legends, Scale Bars and North Arrows in Layouts topic.

Tech Tips

Legends, north arrows and scale bars can be copied and pasted between components. Pasting a legend with the Paste Append command appends the new legend below the existing legend. Copying a legend and pasting it into a map can be a quick way of customizing a map’s legend when a new layer is added.

Using legends with images is a fast way to get RGB color numbers for colors used in icons, web pages and so on. Copy a screen shot to the Windows clipboard with the Print Scrn button on the keyboard, switch to Manifold and Paste As an image into the project pane. Open the image and show a legend to see the colors.
In the example above we have made a screen shot of the Windows desktop, zoomed into the Manifold icon, selected the region about the icon, cropped the image and then displayed a legend. The Manifold icon uses only a few colors. Other icons using 256 colors will require longer legends, of course.

See Also

See the Adding Legends topic for an introduction to legends and a simple example.

See Adding a Legend for an example of creating a legend and then customizing it to change display styles, to use multiple columns and to use a horizontal style instead of the default vertical column style.

See the Thematic Formatting topic for more information on coloring drawings using database attributes.
Legend Text Escape Sequences

Text items in legends may contain values in square brackets [ ] that will be filled in with automatically-generated text whenever the legend display is refreshed.

Escape sequences supported by all legend items:

- [Component] Component name.
- [Computer] Windows name of the computer.
- [Coordinate System] The coordinate system of the component (same as [Projection]).
- [Date] Current date in short date format specified in Windows Regional Options. A typical Windows default is 8/27/2003.
- [Datum] Datum used in the coordinate system of the component.
- [Day] Current day in the month, from 1 to 31.
- [Description] Component description (accessible through the Properties dialog for the component, via the context menu for the component in the project pane).
- [Hour] Current hour, from 1 to 24.
- [Filename] Name plus extension of the project file. In a new project, will be empty until the project is first saved.
- [File Path] Fully qualified name of the project file. In a new project, will be empty until the project is first saved.
- [File Title] Title of the project (displayed in the main window title). Before a project is saved for the first time will be a system generated name such as Project1. After the project is saved, will be the same as the project file name without the .map extension.
- [Long Date] Current date in long date format specified in Windows Regional Options. A typical Windows default is Wednesday, August 27, 2003.
- [Minute] Current minute, from 1 to 60.
- [Month] Current month, from 1 to 12.
- [Month Abb] Current month as a three-letter abbreviation, such as Aug.
- [Month Name] Current month name, such as August.
- [Projection] The coordinate system of the component (same as [Coordinate System]).
- [Second] Current second, from 1 to 60.
- [Time] Current time in format specified in Windows Regional Options. A typical Windows default is 3:41:08 PM.
- [Unit] Unit used in the coordinate system of the component.
- [User] User login running Manifold.
- [Weekday Abb] Current weekday as a three-letter abbreviation, such as Wed.
- [Weekday Name] Current weekday name, such as Wednesday.
- [Year] Current year.

Escape sequences for area, label, line or point formats:
[Column] Name of the column used in thematic format.

[Type] Object type name, such as Areas, Lines, Points or Labels.

[Type / Column] Object type name, with name of the column used in thematic format, if any.

See Also

Adding Legends
Legends
Tables

Tables show data organized into rows and columns. Every row in a table is a record. Every column in a table is a field. We will use the words “row” and “record” interchangeably. We will also use the words “column” and “field” interchangeably.

### Mexico Table

<table>
<thead>
<tr>
<th>ID</th>
<th>Place name</th>
</tr>
</thead>
<tbody>
<tr>
<td>27</td>
<td>AQUIASCALIENTES</td>
</tr>
<tr>
<td>3</td>
<td>BAJA CALIFORNIA</td>
</tr>
<tr>
<td>9</td>
<td>BAJA CALIFORNIA SUR</td>
</tr>
<tr>
<td>21</td>
<td>CAMPECHE</td>
</tr>
<tr>
<td>31</td>
<td>CHIAPAS</td>
</tr>
<tr>
<td>12</td>
<td>CHIHUAHUA</td>
</tr>
<tr>
<td>11</td>
<td>COAHUILA</td>
</tr>
</tbody>
</table>

**Tables and Drawings**

Every drawing in Manifold has a table associated with it. Each object in the drawing (that is, every point, line or area object in the drawing) is linked to a row in the table. Drawings have a table even if there is no data saved for each object. In that case, the drawing's table is mostly empty. It has only one field, the object ID field, for each row. The object ID field is used to link each row to its associated object and cannot be edited. It is indicated as the key field with a small key icon in the column name.

Deleting an object in a drawing will delete the object's record in the drawing's table. Deleting a record in a drawing's table will delete the associated object in the drawing.

**Other Tables**

Tables may also be database tables that are imported into a Manifold project or linked into the project from an external database source. New tables may also be created with Manifold.

Tables may be brought into projects from almost any database, even from those that have nothing to do with drawings or maps. We will often work with tables that have no geographic context. We may wish to prepare data for later use together with drawings and maps, or we might simply wish to use Manifold as a general-purpose means of viewing, analyzing, exploring and managing databases.

The illustration above, for example, shows a project with tables imported from the Microsoft Northwind Traders example database shipped with Microsoft Access.
Manifold® System Release 8.00 User Manual

If we open the Customers table we see the same data familiar to many Access users who have worked with this sample database.

### Tables and Queries

Both tables and queries appear as tables when opened in a table window. Manifold has several types of tables that may appear in a project and that are created in different ways.

<table>
<thead>
<tr>
<th>Type</th>
<th>How Created</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ordinary Tables</td>
<td>Created using File - Create - Table.</td>
</tr>
<tr>
<td>Tables linked to drawings.</td>
<td>Created whenever the drawing is imported or created. Each object in the drawing will be automatically linked to one row in the table.</td>
</tr>
<tr>
<td>Tables that have been imported into the project.</td>
<td>Created using File - Import - Table from a file in one of the database formats supported by Microsoft Data Access Components (CSV, DBF, MDB, XLS, etc) or an arbitrary data source using the Data Sources dialog. Importing a table brings the entire content of the table into the project.</td>
</tr>
<tr>
<td>Tables that are linked to external files or database providers.</td>
<td>Created using File - Link - Table from a file in one of the database formats supported by Microsoft Data Access Components (CSV, DBF, MDB, XLS, etc) or an arbitrary data source using the Data Sources dialog. Linking a table leaves the data outside of the project where other programs can continue to work with it.</td>
</tr>
<tr>
<td>Queries</td>
<td>Created using File - Create - Query. Right clicking the query in the Project pane and selecting Open or double clicking the query edits its text. Right clicking the query and selecting Run runs it and displays the resulting records as a table. Action queries do not return any records. Queries can reference any table or query in the project, including tables linked from external data sources.</td>
</tr>
</tbody>
</table>

Note that all of the above tables can be opened in table windows and manipulated using essentially identical methods. Tables that are linked to drawings can be opened in their own windows just like any other table, and often are. The only difference between tables linked to drawings and other tables is the dynamic tie between drawings and their tables. Every object in a drawing is represented by a record in the table. A selection made in the drawing appears in the table and vice versa. Deleting an object or a record in the table will immediately take effect in the linked component as well. See Drawings and Tables for examples.

### Importing or Linking Tables

Once a table is imported into a Manifold project the table and the data it contains are managed by the Manifold database engine. However, the process of opening external database files or providers and getting data from them into Manifold is managed with connection technologies embedded within Manifold. When external tables
are linked into a project (instead of being imported) the desired connection technologies will be used to manage access to those tables.

All data access technologies are included even though their capabilities overlap when working with simple files. By providing all methods Manifold provides users with convenience, reliability and flexibility. OLE DB for read/write connections or ADO.NET for read-only connections are recommended for maximum speed.

Jet is the database engine used by Microsoft within Microsoft's own Access database products. The File - Import - Table menu choice uses Jet to open simple file types and to import their tables and data into the Manifold project. Providing Jet within Manifold allows users to open simple database file formats that are frequently encountered in Microsoft Office installations.

With the Import dialog, different formats may be opened by simply choosing the desired database type in the Files of type list as shown above. Using Jet also provides high reliability since this is Microsoft's own code used to open simple Microsoft file types that are widely used throughout Office and similar applications.

Choosing Data Sources in the Files of type list will allow us to connect to a data source using the Data Source dialog, which can create a data source with whatever other drivers that are on our system, such as ODBC or OLE DB drivers. Most databases, especially older types such as Btrieve, may have ODBC drivers even if they have not been updated with OLE DB drivers.

OLE DB is a newer, faster, more flexible technology that allows general connection to any database system for which there is an OLE DB provider. OLE DB has now superseded the older ODBC methodology.

OLE DB or ODBC provide great flexibility, since they are methods of connecting to any database file or provider for which we have an OLE DB or ODBC driver on our system. If we install Microsoft upgrades or new Microsoft products on our computer that install newer OLE DB providers, Manifold automatically will be able to take advantage of the new providers.

Microsoft Data Access Components includes the standard suite of Microsoft OLE DB providers, as can be seen in the dialogs used to create a new data source within the Data Source dialog.
There is one OLE DB provider of special interest for legacy database access. The Jet 4.0 OLE DB Provider essentially duplicates the capabilities of Manifold's built-in copy of Jet. Using it is equivalent to using Jet via the File - Import - Table menu choice.

ADO .NET is an even newer database access technology than OLE DB and is a standard means of accessing data for applications using the Microsoft .NET Framework. ADO .NET is known for its simplicity and performance; however, it does have the limitation that ADO .NET is generally read-only.

**Importing vs. Linking Tables**

Any of the supported connection methods can be used to import tables and their data into projects or to link to tables for which the data remains in external files or providers. Once a table is imported into a project the Manifold database engine takes over all management of the data. Therefore, once a table is imported there is no difference between tables regardless of what connection methodology was used for the import.

However, if tables are linked into a project and their data remains in external files or providers, then whichever method was used for the linkage will continue to be used to access the tables and the data they contain. In addition, performance and other characteristics will be subject to the advantages or limitations of whatever file format or provider is being used.

**Linking to External Tables and Multi-User Applications**

When linking to an external file or provider used by multi-user applications Manifold will participate correctly in any multi-user accesses that might go on simultaneously from other applications. This is true for all types of connections.

For example, if we link to a table in an external SQL Server database some other person or process could simultaneously open that table with SQL Server over our corporate network as well. They could access the table from a different machine and work on it from that other machine even as we continue to work on the same table with Manifold. Any updates or changes done by Manifold will be consistent with any locking or other protocols designed to maintain order in multi-user applications. This allows Manifold users to work with external databases from within Manifold even as they are being used by different applications.

Choose the data access method carefully when embarking on demanding multi-user applications. For example, although Jet may be used for multi-user operations with .mdb Access files it is not as robust and reliable a system for multi-user work as is SQL Server. Manifold users can always use the SQL Server Express installation available on the Manifold CD to create highly robust multi-user capable database tables.

**Linking Drawing Data Attributes to External Tables**

So far we've seen that every drawing is linked to a table and that the drawing's table is created within the project at the same time the drawing is created. All such tables exist entirely within the project. Sometimes we would like to link our drawing to an external table that is linked into the project.

Perhaps, for example, we have a drawing of European countries and we would like to color those countries with a thematic format that uses our company's sales for each country. If the company's sales are saved in a table in some corporate server it would be nice to link that external table to the drawing so that every time we open the drawing window the colors in the thematic format are taken from the actual sales recorded on that moment in the corporate server.

To do this we use a relation to link the drawing's table with an external table. See the Attaching External Tables to Drawings topic.

**Tips**

- The fastest performance for smaller tables is achieved by importing a table into a Manifold project so the data it contains may be managed directly by Manifold's own database engine. Compared to big-time DBMS packages like Oracle or SQL Server there is no advantage for medium or large tables, and in the case of very large tables it will often be faster to use an external, enterprise-class DBMS like Oracle or SQL Server.
- Tables that must be shared with other applications should be linked into the project so that their actual data remains outside Manifold where it is easy for other applications to reach the data without needing to know anything about Manifold.
• Use the most direct access method. For example, if the data is in Excel .xls files, open them directly using Jet via File - Import - Table. Don’t use the OLE DB Provider for ODBC Drivers to open an Access ODBC driver that is then used to open the .xls file!
• Tables in Manifold are limited to 4 gigabytes of size. Tables linked from external DBMS packages can be unlimited in size.
• Learn to use the Transform toolbar for fast edits and management of tables.
• Read the Queries topics as well as all the topics in the tables Help book.
• Modern Windows edition (2000 and subsequent) users can drag and drop columns in tables to move them left / right.
• By default, table windows do not use horizontal or vertical gridlines between rows and columns. Gridlines may be turned on in Tools - Options.
• From the context menu for table columns (right click onto a table column head) choose Best Fit, Best Fit All or Best Fit Titles to set width of table columns automatically.
• Manifold tables stored within a project do not support NULL values; however, tables or queries that are linked into a project will support NULL values if their originating data provider supports NULL values. Such columns are said to be NULL-able.

See Also

• See the Working with Tables topic for an introduction to table controls, highlighting methods and other user interface issues.
• ViewBots are important analytic instruments often used with tables. See the ViewBots topic for examples of ViewBot usage.
• Forms and their associated scripts are important tools when customizing operations involving tables.
• Hyperlinks for examples of using URL columns in tables to launch browser sessions when an object in a drawing is double-clicked.
• See the Data Source dialog topic for information on connecting to data sources more sophisticated than simple file types.

Examples Used

Many of the illustrations for tables are screen shots made using the Nwind.mdb sample database provided by Microsoft. This is a version of the Northwind.mdb sample database distributed with Access and thus familiar to many Microsoft Office users. Nwind.mdb is provided on the Manifold System CD as a sample.
**Importing and Linking Tables**

Tables in Manifold projects may come from a variety of sources. They may be imported into the project or left outside the project and linked into the project. In addition, when either importing or linking a table into a project we have a choice of opening common file types (such as MDB or XLS) directly, or using the Data Source dialog to connect to databases using common technologies such as OLE DB, ODBC, ADO.NET or selected native DBMS connection technologies.

Except for direct access via the Oracle Call Interface to Oracle databases (see below), ADO.NET is usually the fastest way to connect, although it allows only read-only access to tables. OLE DB is a general means of accessing almost any type of provider that is a more modern and efficient method than ODBC, an earlier technology intended to provide universal access to different databases. In comparison to ADO.NET, ODBC is dramatically slower. Manifold supports ADO.NET, OLE DB and ODBC since different databases may have either OLE DB or ODBC drivers.

To import or link a table into a Manifold project

1. Choose File - Import - Table or File - Link - Table depending on whether you wish to import the table or to link it.
2. In the Files of type box choose the access method desired, either choosing a common file type such as .MDB for direct connection or choosing Data Sources () to connect via the Data Source dialog.
3. When using a common file type, browse over to the database file and open it. When using the Data Source dialog, double-click on the data source to open it if it has already been added to the list in the Data Source dialog. Create the data source if it has not yet been added, as is shown in the Data Source dialog topic.
4. Choose the table to be imported or linked.
5. When linking tables, check Read Only if you wish a read-only link.
6. Press OK.

After the initial choice of File - Import or File - Link, all dialogs are the same. The only difference is whether the data in the table is copied into the project or if it is left outside in an external file or data source.

Alternate method to import or link a table

1. Open the Tools - Database Console dialog.
2. In the Data source box, specify a data source by pressing the [...] browse button and then choosing a data source from the list in the Data Sources dialog.
3. In Database Console, press the Refresh button to display the contents of the data source.
4. The upper pane will show the contents of that data source, including any tables and the fields they contain. Click on a table to highlight it.
5. Click on the Import button in the dialog's toolbar to import the highlighted table. Click on the Link button to link the table.

In all cases, tables that are imported or linked will appear to be part of the project like any other table. Manifold can import data from or link to a very wide range of database files and database providers. Manifold supports use of schemas when importing or linking data from SQL Server, Oracle or similar data sources.

**Importing vs. Linking**

- **Importing** a table will bring a copy of that table's data into the Manifold project file.
- **Linking** a table will leave the table's data in a file or data source outside the Manifold project.

Both methods have advantages and disadvantages depending on how the project will be used.

**Advantages of Importing**

- **Speed** - Imported tables will provide much faster performance than using an external data source that is a slow file format (such as a .csv text file or an Excel .xls file); however, tables linked from fast external data sources, such as Oracle or SQL Server will be faster for very large tables.

- **Convenience** - When tables are imported into the project, copying the .map project will automatically copy all tables in the project as well. There is no need to worry about tables that may be resident in other files or maintaining links to database systems that may be inaccessible from different systems.

- **Control** - Importing the data provides complete control over the data and assures that no external program will alter it in a way that is incompatible with its use within the project. If other programs could alter the table (if it were linked and not imported) they might alter it in a way that is incompatible with the Manifold project, for example by removing a field that is used elsewhere within the project.

### Disadvantages of Importing

- **Size Limitation** - Manifold cannot import tables that exceed four gigabytes in size. This limitation does not apply to external, linked tables.

- **Single User Tables** - Only one user can work with Manifold at a time, so that once a table is imported into Manifold it is not accessible by other programs in a multi-user way. An exception is using the Manifold ODBC interface to connect to a Manifold project from a different Manifold session. However, if that's what we have in mind it is usually more conceptually easier to remember what is going on by simply storing shared data in an external database, such as SQL Server Express or Oracle Express.

### Advantages / Disadvantages of Linking

- **External Modifiability** - The main advantage of linking tables is that it leaves the data in an external file or data source where other programs can easily modify it. Of course, if we don't want to risk uncontrolled modifications this is a disadvantage.

- **One Copy of Data** - Linking tables avoids making copies of potentially very large data sets. When a database is a standard archival source, always linking to it instead of copying it assures that the version in use in the project is the currently archived version. The disadvantage, of course, is that one must remember to make copies of linked tables when interchanging projects. Also, changes made in Manifold operations can result in unwanted alteration of the archival original.

- **Multi User Operations** - When linking to external tables, Manifold System is fully multi-user capable and can simultaneously work with, for example, an SQL Server or Oracle table on a remote server even as other users continue to work with the same database. If we are looking at an external table in which data is unexpectedly changing, we must keep in mind the possibility that users on other machines are also working on that same table at the same time and are making changes to the data it contains.

- **Larger Tables** - External (linked) tables can be much larger than the four gigabyte limit on internal (imported) tables.

- **NULLs Supported in Linked Tables** - Although native tables in Manifold do not allow use of NULL values, linked tables may use NULLs if the data source to which they are linked allows the use of NULL values. Such columns are said to be **NULL-able**.

### Important Note when using 64-bit Manifold Editions

Due to a lack of required Microsoft facilities in 64-bit Windows systems, Manifold in 64-bit mode cannot export, import, export or link to DB, HTML, MDB, XLS or WKx format files. This includes no access to the MDB parts of Manifold MFD and MapInfo TAB imports. The workaround for importing or exporting such files is to launch Manifold in 32-bit mode by using the Manifold System (32-bit) shortcut, perform the export from or import into a .map project file and then re-launch Manifold in 64-bit mode using the Manifold System (64-bit) shortcut.

Linking is more complex: the data must be in some format usable in 64-bit mode within 64-bit Windows systems, such as a SQL Server database. Alternatively, the data can be kept within a Manifold .map project file and linked using the Manifold ODBC driver.

### Linked Tables are Read / Write

A very useful aspect of linked tables in Manifold is that they are a "live" connection to the data source from which they are linked. We can make changes in the linked table and those changes will be implemented back to the data source from which the table is linked. In particular:

- A linked table allows adding or paste-appending a new column, renaming an existing column, and changing the type of an existing column.
- We can change the design of a linked table.
- It is possible to geocode records in a linked table.
Columns in a linked table work with transfer rules just like native tables.

All of the above capabilities apply to both standalone linked tables and tables bound to linked drawings. In fact, Manifold's ability to manipulate linked tables is so extensive that Manifold is often used as a front end for databases having nothing to do with GIS, to take advantage of the Transform toolbar and other facilities.

For example, if we would like to add a primary key field to a table in an Access .mdb database, we can launch the Database Console and browse to the .mdb, the browse to the table desired and use the Database console toolbar Link button to instantly link the table into our project.

Open the linked table. Choose Table - Design to see and alter the design of the table. Add a new field to be the key field using a default 32-bit integer type. In the Table Design dialog toolbar press in the View Extended Properties mode button. In the Unique column that thus appears for the key field, choose primary key as the value. Press OK.

Although we do the above work on a linked table in the Manifold project, because the linked table is a direct visualization of the table in the Access .mdb database, the changes we make to the linked table instantly take effect in the Access .mdb as well.

Unlinking a Table

A table that has been linked into a project can be converted into an unlinked table, as though it had been imported instead of linked. To do so, right click on the table in the project pane and choose Unlink. This will import a local copy of the table into the project and will eliminate the link.

Relinking a Table

A linked table may be disconnected from its data source. For example, a data source might be resident on a different machine that crashes. Use the Relink command (in the Table menu and in the context menu when clicking on a table in the project pane) to reconnect the linked table to a data source.

For example, one might create a .map file that contains a linked table that is linked to an Access .mdb file. If the .map file and the .mdb file are moved to another machine the link might not continue to work. In that case one can use the Relink command to reconnect the table in the project to the .mdb file.

Properties

We can get useful information on a linked table by right clicking on it in the project pane and choosing Properties to see the View - Properties dialog. Clicking the [...] browse button to the right of the Status information line opens a very useful dialog that shows the data source of a linked component, the link technology and the name of the rowset, if the component is linked from a table or query and the behavior of the data in multi-user editing scenarios.

Cascading Links and Refresh Data

Manifold allows cascading links, where one linked component is created from another linked component. Circular links are resolved automatically. For example, suppose we create a linked table called Table A in a project that is linked from a geocoded table called Table X in an external data source, such as a table in a SQL Server or Oracle database. We could then create a linked drawing in the project that is linked from that linked table. Let's say we call that linked drawing Drawing A.

However, when a cascading link is created in this way, refreshing the linked drawing will update the drawing to any changes made in the linked table, but it will not cause a cascading refresh backwards out of Manifold to the external database. That is, if we make any changes in Table A and we then refresh Drawing A, the drawing will show those changes. However, if we make any changes in the external Table X and then we refresh Drawing A, the changes will not appear in the drawing. We must first refresh Table A so that it incorporates the latest changes in the external Table X and then we can refresh Drawing A.

Preserving Paths to Linked Tables
Linked tables are saved within Manifold projects using relative pathnames. Moving a .map file that contains a linked table will preserve the link to the table provided that the relative path from the .map file to the linked table file has not changed.

For example, suppose we have a .map file located in C:\projects\Carson\mymap.map that contains a linked table located in the file C:\projects\Carson\tables\airport.dbf. If we move the .map file to D:\backups\January\mymap.map it will still preserve the link to the table if the table file is moved to D:\backups\January\tables\airport.dbf.

ADO.NET, OLE DB, ODBC and Common File Types

Both the File - Import - Table and the File - Link - Table dialogs allow opening a wide range of file types. In addition to common file types such as Access MDB and Excel XLS, Manifold can also open "file types" using ADO.NET, OLE DB or ODBC data sources. Choosing OLE DB or ODBC uses Microsoft OLE DB or ODBC drivers to connect to a data source. Choosing a common file type uses a direct connection via Microsoft's Jet database engine to import or to link to file formats understood by Jet.

Common file types allow import tables from CSV (comma separated values ASCII files), DB (Borland Paradox), DBF (dBase and FoxPro), HTML (tables in web pages), MDB (Access 2000 and Access 97), UDL (Microsoft Universal Data Link), WKx (Lotus) and XLS (Excel).

Using the Data Sources dialog allows us to connect using virtually any technology, such as OLE DB providers or ODBC drivers or ADO.NET or ODBC.NET providers as well.

Using Tools - Database Console launches the Database Console for interactive browsing of data sources to see what tables are available and what fields and type of fields they contain. Toolbar buttons allow importing or linking tables from data sources as can be done via the File - Import and File - Link commands. The Database Console dialog also allows interactive execution from within Manifold of command language statements to manipulate the external database.

See the discussion in the Data Source dialog topic regarding various connection technologies.

See the Tables topic for more discussion of direct connections to common file types as compared to using OLE DB or ODBC.

Files of Type Choices

We can specify many different types of data sources for importing or linking tables. Manifold's internal copy of the Microsoft Jet database engine (available when running in 32-bit mode: Jet is not available in 64-bit mode) can directly read database files in the following formats, including all the various sub-variations normally found of these. In addition, we can choose to connect using the Data Sources () option to invoke the Data Sources dialog. This will allow us to connect to more sophisticated data sources using connection technologies such as OLE DB, DSN or UDL.

<table>
<thead>
<tr>
<th>Files of type:</th>
<th>CSV Files (*.csv)</th>
<th>DB Files (*.db)</th>
<th>DBF Files (*.dbf)</th>
<th>HTML Files (*.html)</th>
<th>WKx Files (*.wk)</th>
<th>XLS Files (*.xls)</th>
<th>Data Sources ()</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Files of type:</th>
<th>CSV Files (*.csv)</th>
<th>DB Files (*.db)</th>
<th>DBF Files (*.dbf)</th>
<th>HTML Files (*.html)</th>
<th>WKx Files (*.wk)</th>
<th>XLS Files (*.xls)</th>
<th>Data Sources ()</th>
</tr>
</thead>
</table>

We can specify many different types of data sources for importing or linking tables. Manifold's internal copy of the Microsoft Jet database engine (available when running in 32-bit mode: Jet is not available in 64-bit mode) can directly read database files in the following formats, including all the various sub-variations normally found of these. In addition, we can choose to connect using the Data Sources () option to invoke the Data Sources dialog. This will allow us to connect to more sophisticated data sources using connection technologies such as OLE DB, DSN or UDL.
No other database software in addition to Manifold System is required to read such files.

For example, virtually all .dbf format files common to dBase and similar systems can be directly read. When text file types (.txt, .asc) are opened, Manifold will open sub-dialogs as necessary to allow user assisted parsing of the text file format. Note that import of text table follows usual PC industry conventions. For example, the first line of the text table should contain the names of the fields in the table.

Choosing Multiple Tables

When using File - Import - Table or File - Link - Table we can choose to import or link more than one table at a time from a database file that contains multiple tables.

For example, if we wish to import tables from Microsoft's sample Nwind.mdb database, we choose several at a time.

Click on any table to highlight it. Click on more tables using CTRL click or SHIFT click in the usual Windows way to highlight additional tables.

SHIFT clicking on another table will highlight all the tables in between.

Note: Some databases have many tables. To see the tables in sorted order, click on the Name column head in the dialog.

Windows Locale Settings and Table Import

Manifold's table import routines will recognize Windows locale settings for .mdb, .xls, some .dbf files (depending on drivers used) and some other formats and will automatically convert text fields to Unicode when the database file locale is incompatible with current regional options. For example, an .mdb database created with Spanish locale settings will have text fields automatically converted to Unicode if it is imported on a Windows machine employing French settings.
Effect of Tools - Options Settings

When importing tables, keep in mind that the default setting in Tools - Options is to not import empty columns. Check the Import empty columns in tables option to import tables with columns that do not contain any values in the originating table.

Note also the setting of Trim strings imported from external databases (on by default). This will delete leading and trailing token separator characters from imported table strings.

The Trim strings option (just like the Trim table transform operators) removes characters listed in the Tool Properties pane’s list of token separators. By default, these are the "white space" characters consisting of the space character, tab, newline and carriage return. Note that adding any other characters to the separator list in Tool Properties will subject them to removal as well if they occur as leading or trailing characters.

Linked Tables and Disconnects

One risk with using linked tables is that the data source might not be available when we want to use the table within our project. For example, the linked table may reside within a database server on a different machine that might experience a hardware crash. To deal with such situations, Manifold uses a conservative strategy when linked tables cannot reach their data sources.

Linked tables that can not reach their data sources will not break any operations in which they participate. Instead, the linked table absorbs the error returned by the database access layer and will configure itself with the same columns that were in use the last time it was opened and the loading process will continue. As a result, occasional crashes of a database server will not break relations between the linked table and other tables in the project. Columns within the linked table will be filled with default values until the server comes back to life and we can reconnect to the server using the Refresh Data command.

One risk of this “fail safe” strategy for dealing with disconnects is that it is possible for a linked table to exist within a Manifold project after some external agency (such as an unwary database administrator) has eliminated the table from its data source. In such cases we will have to delete the linked table manually from the project. Manifold will not automatically delete such tables because in such (hopefully rare) circumstances it is better to have to manually delete in order to be able to take any necessary clean up measures should the linked table have been used in any relations, scripts or elsewhere in the project.

Properties

The View - Properties dialog for components shows the data source of a linked component (such as a linked table) and other relevant information.

The Link / Share dialog accessed from the [...] browse button for linked or shared components accessed from the View - Properties dialog will provide a summary of the link or share properties. This will show the status of a shared component and whether or not changes in a linked component will propagate back to the data source.

Enterprise Edition

If you have installed the Enterprise Edition of Manifold System you will have an additional, very powerful means of importing and linking tables. Enterprise Edition allows centralized storage of components, including tables, within Enterprise servers that are hosted by external database systems, such as DB2, Oracle or SQL Server. Tables may be imported from an Enterprise server or linked into the project while remaining resident on the Enterprise server.

See the Enterprise Edition topic for more information.

See Also

Database Console
View - Refresh Data
Table Window Menus and Controls

When we click onto a table window, Manifold will automatically change menus and toolbars so they contain those menus and controls that work with tables.

Edit, View and Table Menu

**Edit**

- **Undo**
  - Undo last command.

- **Redo**
  - Redo last undone command.

- **Cut**
  - Windows clipboard cut operation.

- **Copy**
  - Windows clipboard copy operation. Use this to "paste" a column's worth of field data into a different column of compatible type.

- **Paste**
  - Paste the contents of the Windows clipboard into this column.

- **Paste Append**
  - Append the contents of the Windows clipboard onto the bottom of this column.

- **Delete**
  - Permanently delete this field from the Table.

- **Delete All**
  - Permanently delete all records from the Table.

- **Select All**
  - Select all records in the table.

- **Select None**
  - Deselect all records in the table.

- **Select Inverse**
  - Invert selection: those records that were highlighted will be deselected while those records that were not highlighted will be selected.

- **Select Mode**
  - Choose the selection mode to be used for mouse selection:
    - **Replace** - Any selection made with the mouse will replace the previous selection.
    - **Add** - Any selection made with the mouse will be added to the previous selection.
    - **Subtract** - Any selection made with the mouse will be subtracted from the previous selection.
    - **Invert** - Any selection made with the mouse will be inverted with the previous selection.
    - **Intersect** - Any selection made with the mouse will be intersected with the previous selection.

- **Find**
  - Find a string or value using various options.

- **Replace**
  - General search and replace dialog.

- **Bookmarks**
  - Toggle (create or clear) a bookmark at the current record, Go to previous or next bookmark or Remove All bookmarks.

- **Go**
  - Go to First, Previous, Next, Last or Blank (append a new record) records.

- **Go To**
  - Go to a particular record by sequential record number or by object ID.

**View**
**Selection Filter**  Show only selected records. See Selection in Tables.

**Columns**  Show or hide columns and specify their order.

**Sort**  Launch sort dialog to allow multi-level sorts.

**Panes**  Turn on and off Manifold panes.

**Full Screen**  Show table using full screen.

**Refresh Data**  Refresh data in a linked table.

**Properties**  Show this table’s properties.

**Table**

**Open Drawing**  Enabled for tables that are linked to drawings. Opens the drawing associated with this table.

**Open Script**  If a table is created with a script, open the script that is associated with this table.

**Add**  Add a new column, a new Active Column or a new Rank Column to the table.

**Address**  Geocode the table against an address geocoding database. See the Street Address Geocoding topic.

**Match**  Spatially geocode the table using a multilevel match of fields against a drawing.

**Design**  Add or delete fields of various types to this table.

**Relations**  Form relations with other tables that will allow use of columns from those other tables in this table.

**Relink**  Enabled for linked tables. Reconnect the table to its data source when the link is severed.

**Unlink**  Enabled for linked tables. Convert the table from a linked table to a regular table. Equivalent to importing the table.

**Tools**

**Database Console**  Shows the contents of OLE DB data sources, including tables, fields and types of fields. Provides toolbar buttons to quickly import or link tables from an OLE DB data source. Allows use of whatever native command language is used by the data provider. A means of browsing and controlling databases interactively without leaving Manifold.

**Note:** Copying tables or table columns or records will copy data to the Clipboard both in internal Manifold format as well as conventional Windows tab-separated text format, which is handy for pasting into Access, Excel and numerous other Windows programs that use tables.

**Navigation Toolbar**

Clicking on a Table window will adapt the navigation toolbar with buttons relevant to Table window operations:

- **First**  Jump current record pointer to the first record in the Table and scroll as necessary to display it.
Previous  Move current record pointer to the previous record in the Table and scroll as necessary to display it.
Next  Move current record to the next record in the Table and scroll as necessary to display it.
Last  Jump current record pointer to the last record in the Table and scroll as necessary to display it.
Blank  Jump to the end of the Table to append a blank record.
Filter  Filter - Specify the criteria used to show or hide records. (Reserved for future expansion. Currently offers only a selection filter option.)
Sort  Launch sort dialog to allow multi-level sorts.

Selection Toolbar

The selection mode buttons control how a new selection made with a selection tool is combined with any existing selection. One selection mode button is always pushed in. In tables the selection mode regulates how clicking on the record handle selects that record.

Replace Selection  Replace the existing selection with the record clicked.
Add to Selection  Add the clicked record to the existing selection.
Subtract from Selection  Subtract the clicked record from the existing selection.
Invert with Selection  If the clicked record is already selected, deselect it. Otherwise, select the clicked record.
Intersect Selection  Select the clicked record only if it is already in the selection. Deselect anything not clicked. This mode usually only makes sense with SHIFT-clicks between the current record and another record.
Selection Filter  Show only selected records in the table.
Selection Style  Change the selection style in drawings to a less intrusive pattern or color or other style. The pull-down arrow next to the Use Selection Color button opens a menu of possible selection styles. This control is normally only used with tables that have an associated drawing and also have the drawing window open. Without changing context from the table one can change the appearance of selected items in the drawing.
Select None  Deselect all items.

Context Menus

Right clicking onto a column head or record handle or cell pops open a context menu with commands that may be applied. These are called "context" menus because the choices they present depend on the context of the right click.

Column Heads (Right click onto a column name)

Note: Columns are the same as fields. The words "column" and "field" are used interchangeably within Manifold.
Sort Ascending | Sort the table using values in this column so that lower numbers or letters nearer to A are at the top.

Sort Descending | Sort the table using values in this column so that higher numbers or letters nearer to Z are at the top.

Add

Add - Column | Add a new column to this table.

Add - Active Column | Add a column in which values are determined by the results of a script. See Active Columns.

Add - Rank Column | Add a column used to rank records by the results of a Decision Support System query. See Rank Columns.

Edit | Edit the definition of an active or rank column.

Edit Script | Edit code for an active column.

Flatten | Convert to a native (regular) column an active column or rank column or a column brought into the table via a relation.

Flatten All | Convert to native (regular) columns all active columns, rank columns and columns brought into the table via relations.

Recompute | Recompute values in an active column.

Cut | Windows clipboard cut operation. Copy selected records to the clipboard and delete them from the table.

Copy | Windows clipboard copy operation. Copy selected records from this table onto the clipboard.

Paste | Paste the contents of the Windows clipboard into this table, replacing selected records.

Paste Append | Paste the contents of the Windows clipboard into this table, appending them to the table.

Delete | Permanently delete selected records from this table.

Rename | Change the name of this column.

Transfer Rules | Specify how the values in this column should be transferred in operations such as those in the Spatial Overlay dialog.

Format | Specify how values should appear in this column.

Width | Specify the width of this column in pixels.

Hide | Show or hide this column. To find hidden columns, use the View - Columns command in the main menu for tables.

Change Type | Change the field type of this column. Note that not all field types may be freely converted. For example, the number 132 can be converted into a text string containing the characters “132”, but the text value “Madrid” cannot be converted into a number.

Language | Specify Windows code page to be used for this column. Different columns in the same table can use different code pages to support different languages. Enabled for ANSI text fields only.
<table>
<thead>
<tr>
<th>Order</th>
<th>Move the column to the top (leftmost position), up, down, or to the bottom (rightmost position) in the display.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best Fit</td>
<td>Arrange the width of this column so that the values in the database will fit in it without extra space.</td>
</tr>
<tr>
<td>Best Fit All</td>
<td>Arrange the width of all columns so that the values in the database will fit in them without extra space.</td>
</tr>
<tr>
<td>Best Fit Titles</td>
<td>Arrange the width of all columns so that the full text of all column head titles is visible.</td>
</tr>
<tr>
<td>Identity</td>
<td>Select the ID column to be used in tables created with queries that have more than one identity column. See the Queries topic.</td>
</tr>
<tr>
<td>Find</td>
<td>Find a particular value in this column. Regular expressions may be used to search for patterns.</td>
</tr>
<tr>
<td>Replace</td>
<td>Find and replace values in this column. Regular expressions may be used to search for patterns and to specify their replacements.</td>
</tr>
</tbody>
</table>

**Records (Right click on record handles at left margin)**

[Note: Cut, Copy, Paste and Paste Append from the Edit Menu work on Records]

| Cut | Windows clipboard cut operation. |
| Copy | Windows clipboard copy operation. |
| Paste | Paste the contents of the Windows clipboard into this record. |
| Paste Append | Append the contents of the Windows clipboard as a new record. |
| Delete | Permanently delete this record from the Table or Query. |
| More Like… | Open a sub-dialog that allows detailed specification of how fields will be used to find additional records that are like this one. |
| More Like This Record | Highlight records that are like this one, considering all the fields that are turned on for display in the Table window. |
| More Like This Record (Sorted) | Highlight records that are like this one, considering all the fields that are turned on for display in the Table window, and sort the database so that the records most like this record are at the top and those least like this record are at the bottom. |
| Coordinates | In tables associated with drawings, show the coordinates that define the drawing object for this record. |

**Cells (Right click onto a cell)**

| Cut | Windows clipboard cut operation. |
| Copy | Windows clipboard copy operation. |
**Paste** Paste the contents of the Windows clipboard into this cell.

**Paste Append** Append the contents of the Windows clipboard to this cell. Works with text types only, and is disabled when cells contain other types.

**Delete** Permanently delete the content of this cell from the Table or Query and restore the default value. For example, using **Delete** on an integer field will reset the value to zero.

**More Like…** Open a sub-dialog that allows detailed specification of how this field will be used to find additional records that are like this one.

**More Like This Cell** Highlight records that are like this one, considering only this field.

**More Like This Cell (Sorted)** Highlight records that are like this one, considering only this field, and sort the database so that the records most like this record are at the top and those least like this record are at the bottom.
Creating New Tables

Manifold can create new database tables. Use the File - Create - Table command to create a new table in a Manifold Project.

To create a new table in a Manifold project

1. Choose File - Create - Table
2. In the Create Table dialog, click the Add Column button to add a new column.
3. Double-click into the Name cell to specify the name for the column.
4. Double-click into the Type cell to specify the type of each column. Click into any other cell to accept the type chosen from the list box.
5. Double-click into the Size cell to specify the size of each column for those field types that allow specifiable size.
6. Repeat steps 2 through 5 for each field that is to appear in the table.
7. Click on any field to highlight it and choose Move Up or Move Down to change the order in which it appears. The top field will be leftmost in the table.
8. When finished, press OK to create the table.

See the Field Types in Tables topic for a complete list of field types supported within Manifold tables.

Tables may also be altered at any time using the Table - Design dialog.

Copy and Paste As

Tables may be created from other components by using Edit - Copy to copy the other component in the project pane and then using Edit - Paste As to paste the component as a table.

Copying tables or table columns or records will copy data to the Clipboard both in internal Manifold format as well as conventional Windows tab-separated text format, which is handy for pasting into Access, Excel and numerous other Windows programs that use tables.

Example: Copy an Image into a Table

1. Click on the project pane to make it active.
2. Right-click on the image component to be copied and choose Copy from the context menu.
3. Right-click anywhere in the project pane other than onto an existing component and choose Paste As... and then Table in the context menus that appear.
4. Choose the options desired in the dialog that appears for how the image data should appear in the table. For example, for each pixel we can create separate fields giving the Red, Green and Blue channel values for that pixel.
5. This creates a new table component where each record in the table is taken from one pixel in the image. The table is "geocoded" with the pixel locations of each pixel in the image together with the values of the pixels.
For more sophisticated use, it is also usually possible to **Copy** from within a component window and to **Paste As**... the data directly into a new component in the project. For example, we can select points within a drawing and then **Copy** those points, click onto the project pane to move the focus there and then use **Paste As**... a table to create a table that contains geocoded records for each copied point.

**Autoscope**

The "scope" of a **Copy** is what is copied. When copying from an open drawing, image or other window, if a selection is present only those items (image pixels, drawing objects, table records, etc) that are selected will be copied. This is an example of **autoscope**, where the scope of the copy is automatically restricted to the selection if one is present.

In contrast, when using **Copy** to copy components that are highlighted in the project pane the entire component will be copied whether or not there is a selection within that component.

**Typical Copy and Paste As Table Combinations**

- **Drawing to Table**: Reads the coordinate locations of points and pastes them as records into a geocoded table. This is a fast way of creating a list of latitude / longitude locations for points in a map, such as a city.
- **Table to Drawing**: Paste a geocoded table as a drawing of points.
- **Image to Table**: Create a table where each record is a "geocoded" pixel from the image.
- **Table to Image**: Create an image from a table where each record represents the pixel coordinates and values for each pixel in the image.
- **Surface to Table**: Create a geocoded table where each record contains an elevation value for a given point in the surface.
- **Table to Surface**: Create a surface taking elevation values for point records specified in a geocoded table.

**Pasting Drawings as Tables**

Pasting a drawing as a table displays a dialog to select which columns to paste and then creates a table with the selected columns. The list of columns includes intrinsic columns as well as any active or rank columns or columns brought via a relation that may be defined in the drawing's table. The resulting table will have a record for each copied drawing object.

For a list of intrinsic columns, see the Intrinsic Fields in Tables topic.

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Latitude</td>
<td>Location of the centroid Y coordinate (Latitude) for the object.</td>
</tr>
<tr>
<td>Longitude</td>
<td>Location of the centroid X coordinate (Longitude) for the object.</td>
</tr>
<tr>
<td>MinimumX</td>
<td>The minimum X or Longitude value of the minimum enclosing box of the object.</td>
</tr>
<tr>
<td>MinimumY</td>
<td>The minimum Y or Latitude value of the minimum enclosing box of the object.</td>
</tr>
<tr>
<td>MaximumX</td>
<td>The maximum X or Longitude value of the minimum enclosing box of the object.</td>
</tr>
<tr>
<td>MaximumY</td>
<td>The maximum Y or Latitude value of the minimum enclosing box of the object.</td>
</tr>
<tr>
<td>NumberOfBranches</td>
<td>The number of branches in areas or lines that are</td>
</tr>
</tbody>
</table>
multiply-branched objects. Always 1 for points.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NumberOfCoords</td>
<td>The total number of coordinates that define the object. Always 1 for points.</td>
</tr>
<tr>
<td>ObjectType</td>
<td>The type of object.</td>
</tr>
</tbody>
</table>

The above computed field values appear in angle brackets <>. If the drawing's table already contains field names like the above, when the table is pasted they will be incremented so that instead of "MinimumY" there will appear a "MinimumY2" field. The **NumberOf...** fields will be pasted as Int32 integers, the **ObjectType** field is pasted as a variable-length Unicode text field and all the other computed fields will be pasted as Float64 floating point numeric fields.

### Pasting Images and Surfaces as Tables

Pasting an image or a surface as a table displays a dialog to select which columns are to be pasted and then creates a table with the selected columns. The list of columns includes all columns defined in the virtual table for the copied image or surface component. The resulting table will have a record for each copied image or surface pixel.

For a list of columns available in virtual tables for images and surfaces, see the Virtual Tables for Images and Surfaces topic.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index</td>
<td>An index into the palette. Appears as a choice only when pasting palette images.</td>
</tr>
<tr>
<td>Color</td>
<td>A decimal number representing the RGB color. The decimal representation of the six-digit hexadecimal number formed by concatenating the three, two-digit hexadecimal values for R, G and B channels.</td>
</tr>
<tr>
<td>Red</td>
<td>A value from 0 to 255 for the Red channel value.</td>
</tr>
<tr>
<td>Green</td>
<td>A value from 0 to 255 for the Green channel value.</td>
</tr>
<tr>
<td>Blue</td>
<td>A value from 0 to 255 for the Blue channel value.</td>
</tr>
<tr>
<td>Alpha</td>
<td>A value from 0 to 255 for the Alpha channel value.</td>
</tr>
<tr>
<td>Intensity</td>
<td>The intensity of the pixel as a value from 0 to 255.</td>
</tr>
<tr>
<td>Hue</td>
<td>The hue of the pixel as a value from 0 to 255.</td>
</tr>
<tr>
<td>Lightness</td>
<td>The lightness of the pixel as a value from 0 to 255.</td>
</tr>
<tr>
<td>Saturation</td>
<td>The saturation of the pixel as a value from 0 to 255.</td>
</tr>
<tr>
<td>Longitude</td>
<td>Longitude or X coordinate of the pixel.</td>
</tr>
<tr>
<td>Latitude</td>
<td>Latitude or Y coordinate of the pixel.</td>
</tr>
</tbody>
</table>

The above computed field values appear in angle brackets <>; **Longitude** and **Latitude** fields are pasted either as as longitude or latitude types or, if the **Latitude / Longitude coordinates** box is not checked as Float64 floating point numeric fields. The **Color** field is pasted as an Int32 integer. All other fields are pasted as Int8 integers.

**Color** appears as a choice only when pasting palette images; however, the RGB and Alpha values appear even if the image is not an RGB or RGBA image. In this case, they represent the R, G, B and Alpha values for the color of the pixel. Note that even though a color may be taken from a palette, it still can be described as composed of R, G and B values.

### Adding Records to a Table

Tables that are associated with drawings cannot have records added to them from the table window since each record in the table is associated with an object in the drawing. Adding a record means adding an object. This is
done in either a drawing window or a map window (if the drawing is used as a layer in a map) by inserting a new point, line or area object. When the new object is inserted into the drawing a new record for it will appear in the table.

To insert new records into tables that are not associated with drawings, click into cells in the new record row (marked with an asterisk) to edit their values. Press **SHIFT-Enter** to append the new record to the table.

Suppose we start with a table with only two records.

<table>
<thead>
<tr>
<th>First</th>
<th>Last</th>
</tr>
</thead>
<tbody>
<tr>
<td>Groucho</td>
<td>Marx</td>
</tr>
<tr>
<td>John</td>
<td>Lennon</td>
</tr>
</tbody>
</table>

If we click on the first empty cell in the new record row and press **Enter** we can begin entering a new name. In this case, we have typed **Ben**.

Pressing **Enter** will enter that value into that cell. Note that the status icon in the row handle has changed to a small pencil to show we are entering a new record. We can press the right arrow on the keyboard to move to the next cell in the row.

If we press **Enter** we can begin editing that cell. After entering the text **Franklin** we can press **Enter** once more to have the cell accept the new value.

If we are happy with our new record we can press **SHIFT-Enter** to stop editing and to append the new record to the table. Note that the editing pencil icon has been replaced with a current record icon in the usual way. A new record row appears as well.

**Specifying Languages for Columns**
Language specification in text columns varies depending on whether Unicode or ANSI text is used. Unicode text columns can simultaneously contain characters from more than one language. ANSI text columns are usually restricted to one language in addition to (possibly) English. Character data contained in an ANSI text column will be interpreted by the system according to the code page associated with that column.

By default, ANSI text columns use the system code page. If desired, any ANSI text column in a table can be set to use a different code page by right clicking on the column head and choosing **Language**.

![Column Language](image)

**Keyboards Editing**

One may also edit data in Table windows using "keyboard style" data editing. **Double-click** into a field to edit it and then push **Enter**. Move the edit box to another field by using the keyboard cursors. Push **Enter** to open the field for editing, enter the new value, and then hit **Enter** to apply the value.

The **Tab** key will move one cell to the right of the active cell. A **Shift-Tab** will jump one cell to the left of the active cell.

Keyboard editing is a rapid way of moving the cursor through columns of data and making changes as desired.

**See Also**

**Editing Data in Tables**
Intrinsic Fields in Tables

Fields that appear in tables are normally ordinary data fields. These may be a part of the table, or they may appear in a table as a result of a relation. For a drawing’s table, Manifold also can display fields generated by the system for each record. These are fields that describe some system attribute of the object represented by that record, or which are computed from the geometry of the object. Such system-generated fields are called intrinsic fields.

Intrinsic columns are not shown by default in tables. We can use the View - Columns command to show intrinsic fields. Intrinsic fields have an (I) as part of their name in the Columns dialog. Intrinsic fields will also appear in the Object Fields dialog, which may be called from the context menu that pops up when right-clicking on an object in a drawing.

Intrinsic fields are an intrinsic part of the table and cannot be deleted with the Table - Design dialog. They may be used as key fields for relations or for other purposes just like ordinary data fields.

Intrinsic Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area (I)</td>
<td>Object area.</td>
</tr>
<tr>
<td>Bearing (I)</td>
<td>0 for points and areas. For lines, the bearing in degrees of a straight line segment constructed between the first and last points of the very first branch within the line. Computed using ellipsoidal trigonometry. 90 is East and 270 is West.</td>
</tr>
<tr>
<td>Branches (I)</td>
<td>The number of branches (indicates topologically branched objects, such as areas with holes or islands).</td>
</tr>
<tr>
<td>Coordinates (I)</td>
<td>The number of coordinates that define the object.</td>
</tr>
<tr>
<td>Geom (I)</td>
<td>The geometry data (&quot;the object metric&quot;) that defines the position and shape of the object. Read/write. Modifying this column with, say, an UPDATE query, modifies the metric of the associated drawing objects. Setting Geom (I) to a NULL value or to a geometry value the type of which differs from that of the drawing object is not allowed.</td>
</tr>
<tr>
<td>Latitude (I)</td>
<td>Latitude of the center of the object in degrees latitude.</td>
</tr>
<tr>
<td>Length (I)</td>
<td>Object length (or perimeter).</td>
</tr>
<tr>
<td>Longitude (I)</td>
<td>Longitude of the center of the object in degrees longitude.</td>
</tr>
<tr>
<td>Selection (I)</td>
<td>Boolean: currently selected or not.</td>
</tr>
<tr>
<td>Selection Mask (I)</td>
<td>A byte giving the saved selections mask.</td>
</tr>
<tr>
<td>Type (I)</td>
<td>A lookup field for object type: area, line or point.</td>
</tr>
<tr>
<td>Version (I)</td>
<td>Current version of the drawing object. Changing the metric of a drawing object increments the value of the Version (I) column. For a linked drawing which uses a version column to facilitate multi-user editing, the values in the Version (I) column will be the same as the values in the version column.</td>
</tr>
<tr>
<td>X (I)</td>
<td>X coordinate of the center of the object in the coordinate system used by the drawing.</td>
</tr>
<tr>
<td>Y (I)</td>
<td>Y coordinate of the center of the object in the coordinate system used by the drawing.</td>
</tr>
</tbody>
</table>

Columns for intrinsic fields will be shown using a different font color, with a different color being used for read-only intrinsic fields such as the Selection Mask.
The **Geom (I)**, **X (I)**, **Y (I)**, **Latitude (I)**, and **Longitude (I)** fields are editable. Changing these fields will **move** or in the case of **Geom (I)** will **edit** the object. These fields are normally used to move the position of points. They may also be used to move the centers of areas or lines, in which case they result in a change to all coordinates defining the line or area required by the horizontal or vertical displacement of the center. See the Editing Intrinsic Fields in Tables topic and also the Geometry in Tables topic for information about geoms.

The intrinsic fields may be used like other fields in the table. For example, the table may be sorted by clicking on the column head for an intrinsic field. This allows some remarkable uses of tables. For example, one can sort all objects in the table from lowest latitude to highest (that is, from South to North) by simply clicking on the **Latitude (I)** column head. We could then click and shift-click to select objects in a desired range of latitudes.

Intrinsic fields may also be used in SQL queries, Active Column formulas and other uses where an ordinary field might be used. Note that since intrinsic fields have an "(I)" as part of their names they must be used within square brackets in SQL, as in

```sql
SELECT [Latitude (I)] FROM mytable WHERE [Latitude (I)] > 60;
```

The **Geom (I)** field is especially useful for manipulating geometry from queries, such as the following three examples:

```sql
UPDATE [A] SET [Geom (I)] = Buffer([Geom (I)],10);

INSERT INTO B ([Geom (I)]) VALUES (NewPointLatLon(30, 40));

SELECT Centroid([Geom (I)]) FROM C;
```

...where **A**, **B** and **C** are drawing names.

**Native Coordinates**

Native coordinates or native units are numbers stored within the drawing that Manifold uses internally as positional coordinates. These numbers are shown in the Object Coordinates dialog that pops up if you right-click an object and choose **Coordinates**. They are also used in the **X (I)** and **Y (I)** intrinsic field columns.

Projected coordinates are native coordinates adjusted with the local scale and local offset parameters of the coordinate system (projection) in use. In many cases, projected coordinates are the same as native coordinates (because of the local scale and offset values) but sometimes they are different.

The relationship between projected coordinates, native coordinates and the local scale and local offset parameters may be expressed in pseudo-code as:

```
ProjectedX = NativeX * LocalScaleX + LocalOffsetX
ProjectedY = NativeY * LocalScaleY + LocalOffsetY
```

**The Selection Mask**

Each object may participate in the selection and in one or more of any of seven possible saved selections. Whether or not the object is selected or is a member of one of the seven selection sets is coded bitwise into a single byte, the **Selection Mask**.

**Length and Area Units**

The **Area (I)** and **Length (I)** intrinsic fields report the area and length of the object based upon two factors:

- The coordinate system in use: latitude / longitude drawings always use degrees as units while projected drawings will report using either meters or feet.
- The setting of the **Use English measurement units** checkbox in the Tools - Options dialog. This is the same box that specifies units for the Tracker tool. If this checkbox is checked, units for projected
drawings will be reported in feet. If it is not checked, units for projected drawings will be reported in meters.

The above rules also may be summarized as follows:

- If the drawing is in Latitude / Longitude (that is, unprojected) Length (I) is reported in degrees and Area (I) is reported in square degrees.
- If the drawing is projected and the Use English measurement units option is turned off, Length (I) is reported in meters and Area (I) is reported in square meters.
- If the drawing is projected and the Use English measurement units option is turned on, Length (I) is reported in feet and Area (I) is reported in square feet.

For example, projected drawings will use coordinate systems in meters or feet, resulting in Area (I) values being expressed in square meters or square feet depending on the setting of the Use English measurement units checkbox. Unprojected drawings use coordinate systems expressed in degrees, resulting in Area (I) values being expressed in square degrees.

Because the size of a "square degree" varies considerably based on its latitude, values of Area (I) and Length (I) will normally only have acceptable accuracy if the drawing is projected to some projection that provides reasonable accuracy for the region of interest.

**Bearings**

The Bearing (I) field is only accurate when the drawing is projected and only then within that part of the projection within which the accuracy of bearings is preserved. Bearings can only be as accurate as the projection of the drawing allows them to be.

To take an example, consider the case of an unprojected drawing in latitude / longitude. What is a straight line in latitude and longitude coordinates is a curved line in the 3D spherical surface of the Earth. This means that in most cases the value of the bearing will be different at different positions along the line. In cases where the value of the bearing changes along the line, Bearing (I) returns the value of the bearing near the starting location. This can be quite different from the value of the bearing near the end of the line.

To avoid such problems, project the drawing into some suitable projection that preserves angles in the area of interest before using Bearing (I).

Note that the values reported by the Bearing (I) field are ellipsoidal bearings (computed over a 3D Earth ellipsoid), while the values used by the dialog mode of visual editing and selection tools are Euclidean bearings ("flat" bearings within whatever projection is in use). Ellipsoidal and Euclidean bearings are not the same thing and should not generally be used interchangeably. Euclidean bearings are only as good as or as inaccurate as whatever projection is used, so if bearings are important it is also important to select a projection that preserves accurate angles within the area of interest. Projections must always be selected to preserve the map characteristics of interest (see the Projections Readings topic).

**Performance**

Showing all the intrinsic fields in a table will slow down table redisplay speed, since some of the intrinsic fields require computation. Intrinsic fields are computed on the fly whenever a table is displayed. Some fields such as Area (I) are computationally very expensive and thus could dramatically slow down performance to a level unacceptable for interactive work in the case of very large drawings when working on slower machines.

One way of avoiding an interactive slow down is to not display the intrinsic fields but only to use them within SQL queries and other relatively non-interactive processes. Even in this case, though, the fields will still be recomputed on the fly so such queries could take a very long time to run when working with large drawings.

To achieve faster performance we can use the Transform toolbar to Copy an intrinsic field such as Area (I) to a new column in the table. This requires a one-time computation. After the Copy is done, we can turn off the intrinsic field and view the new column, which will be a fixed column that is stored data like all of the other columns. The new column will not be dynamically recomputed so if there are any changes made to the areas in the drawing the new column’s values will not change. In addition, the size of the table will increase as stored on disk because the new column is saved in storage and not computed as needed. Despite these minor limitations this technique is fine for use with drawings where the objects don’t change but where the values reported by the intrinsic fields must be rapidly available.
Because some intrinsic fields, for example, Area (I), take a long time to compute, Manifold caches their values whenever possible. The first time we access such a column (say, by invoking the Object Fields dialog, which queries the value of each intrinsic field, even if the intrinsic fields are initially hidden), Manifold will do the necessary computations and will save the resulting values in cache. On subsequent accesses Manifold will not need to compute the intrinsic field but can simply use values already in cache.

### Intrinsic Fields when Pasting Surfaces and Images

When copying a surface or an image and then pasting as a table, the Paste As dialog will offer intrinsic fields computed for each pixel in the surface or image.

For **surfaces**, intrinsic fields available will be:

- **Center Easting (I)**: X coordinate of the center of this pixel in native coordinate system units adjusted with the values of the local scale and local offset parameters.
- **Center Latitude (I)**: Latitude of the center of this pixel in degrees latitude.
- **Center Longitude (I)**: Longitude of the center of this pixel in degrees longitude.
- **Center Northing (I)**: Y coordinate of the center of this pixel in native coordinate system units adjusted with the values of the local scale and local offset parameters.
- **Center X (I)**: X coordinate of the center of this pixel in native coordinate system units.
- **Center Y (I)**: Y coordinate of the center of this pixel in native coordinate system units.
- **Height (I)**: Height value.
- **Invisible (I)**: 0 or 1 to indicate invisible pixels, normally used for missing height values.
- **Latitude (I)**: Latitude of the lower left corner of this pixel in degrees latitude.
- **Longitude (I)**: Longitude of the lower left corner of this pixel in degrees longitude.
- **Easting (I)**: X coordinate of the lower left corner of this pixel in native coordinate system units adjusted with the values of the local scale and local offset parameters.
- **Northing (I)**: Y coordinate of the lower left corner of this pixel in native coordinate system units adjusted with the values of the local scale and local offset parameters.
- **Selection (I)**: Boolean: currently selected or not.
- **Selection Mask (I)**: A byte giving the saved selections mask.
- **X (I)**: X coordinate of the lower left corner of this pixel in native coordinate system units.
- **X Offset (I)**: X position in pixel coordinates from the lower left corner.
- **Y (I)**: Y coordinate of the lower left corner of this pixel in native coordinate system units.
- **Y Offset (I)**: Y position in pixel coordinates from the lower left corner.

For **images**, intrinsic fields will be appropriate to the type of image. Many fields are the same as those available for surfaces.
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Center Easting (I)</td>
<td>X coordinate of the center of this pixel in native coordinate system units adjusted with the values of the local scale and local offset parameters.</td>
</tr>
<tr>
<td>Center Latitude (I)</td>
<td>Latitude of the center of this pixel in degrees latitude.</td>
</tr>
<tr>
<td>Center Longitude (I)</td>
<td>Longitude of the center of this pixel in degrees longitude.</td>
</tr>
<tr>
<td>Center Northing (I)</td>
<td>Y coordinate of the center of this pixel in native coordinate system units adjusted with the values of the local scale and local offset parameters.</td>
</tr>
<tr>
<td>Center X (I)</td>
<td>X coordinate of the center of this pixel in native coordinate system units.</td>
</tr>
<tr>
<td>Center Y (I)</td>
<td>Y coordinate of the center of this pixel in native coordinate system units.</td>
</tr>
<tr>
<td>Invisible (I)</td>
<td>0 or 1 to indicate invisible pixels, normally used for missing height values.</td>
</tr>
<tr>
<td>Latitude (I)</td>
<td>Latitude of the lower left corner of this pixel in degrees latitude.</td>
</tr>
<tr>
<td>Longitude (I)</td>
<td>Longitude of the lower left corner of this pixel in degrees longitude.</td>
</tr>
<tr>
<td>Easting (I)</td>
<td>X coordinate of the lower left corner of this pixel in native coordinate system units adjusted with the values of the local scale and local offset parameters.</td>
</tr>
<tr>
<td>Northing (I)</td>
<td>Y coordinate of the lower left corner of this pixel in native coordinate system units adjusted with the values of the local scale and local offset parameters.</td>
</tr>
<tr>
<td>Selection (I)</td>
<td>Boolean: currently selected or not.</td>
</tr>
<tr>
<td>Selection Mask (I)</td>
<td>A byte giving the saved selections mask.</td>
</tr>
<tr>
<td>X (I)</td>
<td>X coordinate of the lower left corner of this pixel in native coordinate system units.</td>
</tr>
<tr>
<td>X Offset (I)</td>
<td>X position in pixel coordinates from the lower left corner.</td>
</tr>
<tr>
<td>Y (I)</td>
<td>Y coordinate of the lower left corner of this pixel in native coordinate system units.</td>
</tr>
<tr>
<td>Y Offset (I)</td>
<td>Y position in pixel coordinates from the lower left corner.</td>
</tr>
</tbody>
</table>

In addition there are intrinsic fields specific to images:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Color (I)</td>
<td>RGB color value expressed as a decimal number (the decimal equivalent to a hexadecimal color value in the form #rrggb).</td>
</tr>
<tr>
<td>Color Index (I)</td>
<td>A color index from 0 to 255 available for palette images only.</td>
</tr>
<tr>
<td>Red (I)</td>
<td>Red channel value from 0 to 255.</td>
</tr>
<tr>
<td>Green (I)</td>
<td>Green channel value from 0 to 255.</td>
</tr>
<tr>
<td>Blue (I)</td>
<td>Blue channel value from 0 to 255.</td>
</tr>
<tr>
<td>Alpha (I)</td>
<td>Alpha channel value from 0 to 255.</td>
</tr>
<tr>
<td>Intensity (I)</td>
<td>Intensity value from 0 to 255.</td>
</tr>
<tr>
<td>Hue (I)</td>
<td>Hue value from 0 to 255.</td>
</tr>
</tbody>
</table>
Lightness (I)  Lightness value from 0 to 255.

Saturation (I)  Saturation value from 0 to 255.

**Virtual Tables**

Images and surfaces in Manifold have virtual tables that allow queries to access the image using intrinsic fields like the above. See the Virtual Tables for Images and Surfaces topic for details.

**Pixel Coordinates**

The X Offset (I) and Y Offset (I) values give the X and Y coordinates of the pixel in column and row pixel coordinates. If an image is 8192 pixels wide (8192 columns) and 4096 columns high (4096 rows) the X Offset (I) coordinates of the pixels will range from 0 to 8191 and the Y Offset (I) coordinates of the pixels will range from 0 to 4095. The pixel in the upper left of the image (first row, first column) will have coordinates of X Offset (I) = 0 and Y Offset (I) = 0. The pixel in the lower left corner will have coordinates of X Offset (I) = 0 and Y Offset (I) = 4095. The pixel in the lower right corner of the image will have coordinates of X Offset (I) = 8191 and Y Offset (I) = 4095.

Pixels in georegistered images can cover a large region, quite possibly several kilometers in size. The positions reported by X (I), Y (I), Longitude (I) and Latitude (I) report the location of the lower left corner of the pixel.

**Tech Tip**

Note that the Selection (I) intrinsic is a Boolean. It may appear as a 1 or a 0 value or Yes or No or True or False because that is the way the column is formatted for appearance. However, when used in a query, the true Boolean value of Selection (I) should be checked as True or False.

Therefore, Selection (I) = true is a valid test in a SQL query but [Selection (I)] = 1 will not work. It is a coincidence of conversion rules that [Selection (I)] = -1 will also work.

**See Also**

See Editing Intrinsic Fields in Tables for notes on editing intrinsic fields.

See the Add Points with Instant Data example topic for an example of moving points by editing intrinsic fields.
### Field Types in Tables

Manifold supports the following field types in tables:

<table>
<thead>
<tr>
<th>Field Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Binary (fixed-length)</strong></td>
<td>Untyped binary data. Often used for programming masks and coded status words.</td>
</tr>
<tr>
<td><strong>Binary (variable-length)</strong></td>
<td>Untyped binary data.</td>
</tr>
<tr>
<td><strong>Boolean</strong></td>
<td>Yes/No, True/False, On/Off. Standard Microsoft data type.</td>
</tr>
<tr>
<td><strong>Character (ANSI)</strong></td>
<td>A single character. Standard Microsoft data type.</td>
</tr>
<tr>
<td><strong>Character (Unicode)</strong></td>
<td>A single character. Standard Microsoft data type.</td>
</tr>
<tr>
<td><strong>Coordinate System</strong></td>
<td>Coordinate system data in Manifold binary format.</td>
</tr>
<tr>
<td><strong>Currency</strong></td>
<td>Fixed length, eight-byte type used for money values. Eight bytes can specify precise monetary amounts that are greater than all the money there ever has been in the world. Standard Microsoft data type.</td>
</tr>
<tr>
<td><strong>Date and time</strong></td>
<td>Standard Microsoft date/time field type.</td>
</tr>
<tr>
<td><strong>Floating-point (double)</strong></td>
<td>Standard Microsoft data type.</td>
</tr>
<tr>
<td><strong>Floating-point (single)</strong></td>
<td>Standard Microsoft data type.</td>
</tr>
<tr>
<td><strong>Geometry</strong></td>
<td>Coordinate data combined with coordinate system data in Manifold binary format.</td>
</tr>
<tr>
<td><strong>Geometry (SDE)</strong></td>
<td>Coordinate data in ESRI SDE binary format.</td>
</tr>
<tr>
<td><strong>Geometry (SHP)</strong></td>
<td>Coordinate data in ESRI SHP binary format.</td>
</tr>
<tr>
<td><strong>Geometry (WKB)</strong></td>
<td>Coordinate data in OGC WKB binary format.</td>
</tr>
<tr>
<td><strong>Integer (8-bit)</strong></td>
<td>Standard Microsoft data type.</td>
</tr>
<tr>
<td><strong>Integer (8-bit, unsigned)</strong></td>
<td>Standard Microsoft data type.</td>
</tr>
<tr>
<td><strong>Integer (16-bit)</strong></td>
<td>Standard Microsoft data type.</td>
</tr>
<tr>
<td><strong>Integer (16-bit, unsigned)</strong></td>
<td>Standard Microsoft data type.</td>
</tr>
<tr>
<td><strong>Integer (32-bit)</strong></td>
<td>Standard Microsoft data type.</td>
</tr>
<tr>
<td><strong>Integer (32-bit, unsigned)</strong></td>
<td>Standard Microsoft data type.</td>
</tr>
<tr>
<td><strong>Latitude</strong></td>
<td>Manifold double-precision floating point Latitude.</td>
</tr>
<tr>
<td><strong>Longitude</strong></td>
<td>Manifold double-precision floating point Longitude.</td>
</tr>
<tr>
<td><strong>Percentage</strong></td>
<td>Percentage amounts from 0% to 100%. Also used as ranks when reporting Decision Support System query results.</td>
</tr>
<tr>
<td><strong>Text (ANSI, fixed-length)</strong></td>
<td>Standard Microsoft data type.</td>
</tr>
<tr>
<td><strong>Text (ANSI, variable-length)</strong></td>
<td>Standard Microsoft data type.</td>
</tr>
<tr>
<td><strong>Text (Unicode, fixed-length)</strong></td>
<td>Standard Microsoft data type.</td>
</tr>
<tr>
<td><strong>Text (Unicode)</strong></td>
<td>Standard Microsoft data type.</td>
</tr>
</tbody>
</table>
variable-length

**URL**  Internet URL. Can include protocol, server, ports, URL object and query parameters. See the Hyperlinks topic for special uses of URL columns in drawing tables.

Use variable-length types when the amount of data within the field can be varied. Variable length fields have no practical limit in size (they cannot contain data greater than 2 gigabytes). There is effectively no wasted space when variable-length fields are used. We could use them all the time except that often it is more convenient (for the design of dialog boxes or when programming) to use fixed-length fields so that we always know the maximum length of a field.

However, there is no storage penalty for using fixed length fields. Manifold uses a dynamic storage allocation strategy so that there will be no wasted space regardless of whether fixed-length or variable-length text fields are used.

**Lookup Columns**

Lookup columns are a special type of text column that is constrained to a limited set of specific text values. Instead of permitting free entry of text when editing a cell, a pull-down list of acceptable values appears from which a choice may be made.

Creating a lookup column is a three-step process: First, we create a text field in the table of the desired type (fixed or variable length ANSI or Unicode text). Second, we enter records into the table so that there is at least one example in the table of each possible text value that will be allowed. Finally, we change the column type of that text field to `<lookup>`.

To edit allowed values in a lookup table, we first convert the lookup field back into a regular text type field, add one or more new records providing examples of the desired new values, and then convert the field back into a lookup type.

**Example**

Let's create a table that will be used to keep track of inventory of parts for motorcycles. Each part will be painted black, blue, red or green.

We begin by creating a table with three columns. The **Quantity** column is a numeric column and the other two columns are Unicode text columns.
We open the table and add four records to it. Each record has one of the four allowed choices of color in the Color column. What we enter in the other cells doesn’t matter, since we are using these as “dummy” records to populate the Color column with at least one example of each of the allowed values.

Next, we Right-click onto the Color column head and choose Change Type in the context menu.

We change the type of the Color column to [Lookup] and press OK.

We change the type of the Color column to [Lookup] and press OK.
The result is a table that now has a lookup type in the **Color** column that must be one of the four allowed choices. When we add additional records, a combo box will appear allowing us to choose one of those four values.

If we like, we can delete the dummy records in the table and begin adding new records in earnest that actually reflect the inventory we have on hand of different parts in different colors.

**Lookup Column Options**

The **Value** and **Map To** options allow us to specify one to many relationships, so that more than one value that occurs in the table can map to the same value. This makes it possible to reduce tables that contain many allowed values into a smaller number of allowed values.

For example, suppose we have four colors of gas tank in stock as seen above. Due to a styling changes our company will no longer produce motorcycles in red or green so tanks in those colors will be repainted to black or blue with inventory levels adjusted accordingly.
We can launch the **Change Type** dialog again and change the **Map To** value for **Green** to **Black** and also change the **Map To** value for **Red** to **Blue**.

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Description</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Gas Tank</td>
<td>Black</td>
</tr>
<tr>
<td>1e</td>
<td>Gas Tank</td>
<td>Black</td>
</tr>
<tr>
<td>27</td>
<td>Gas Tank</td>
<td>Blue</td>
</tr>
<tr>
<td>36</td>
<td>Gas Tank</td>
<td>Blue</td>
</tr>
</tbody>
</table>

When we press **OK** we will see that all **Green** values in the table will have been changed to **Black** and all **Red** values in the table will have been changed to **Blue**.

**Use with Drawings**

When using lookup columns in tables we first create records in the table containing each of the required values and then we use the **Change Type** dialog to change the type of the column to **[Lookup]**. When working with drawings, if we don’t already have objects in the drawing that contain an example of each of the possible values we want to allow in the lookup column, we will have to add some temporary objects to the drawing with the desired values.

This is easily accomplished by adding points to the drawing with **Instant Data** turned on. Add points representing the range of allowed values for the lookup column, open the drawing’s table and convert the text column to a lookup column and then delete the points that were added.

**See Also**

**Tables**

**Intrinsic Fields in Tables**
### Selection in Tables

Clicking on a record handle in the table is the same as a **Select Touch** on that record.

In the illustration above we click on the record handle for the **Nevada** record to select it. Selecting a record will combine it with any existing selection in accordance with the selection mode (**Replace**, **Add** and so on) currently in force.

If the table is linked to a drawing it will select the object associated with that record as well. When tables are linked to drawings each record in the table corresponds to one object (a point, line or area) in the drawing. The objects in the drawing and the records associated with them become the same thing for all intents and purposes. We can then consider the table as simply an alternate way of looking at the objects in our drawing.

If a table is linked to a drawing, making a selection in the drawing will automatically select the records in the table associated with the selected objects. These records will appear highlighted in red selection color. Likewise, making a selection in a table by highlighting records will select the associated objects in the drawing.

Just like drawings and images, tables also can save selections in the Selections pane. When a table is linked to a drawing, the saved selections will be the same for both the table and the drawing. Showing a preview from the Selections pane will cause those records in the previewed selection to be displayed with light blue background color.

#### Selecting Records in Tables

We select records in tables by clicking on their record handle. Manifold supports the usual Windows "power moves" to highlight items in tables, with slight modifications to deal with selection modes.

- Clicking on a cell in a record moves the focus to that cell without changing the selection.
- Clicking on a record handle selects the record or a range of records in accordance with the current selection mode.
- Holding the **SHIFT** key down while clicking applies the selection operation to a range of records. This is often used to highlight a series of records: click on the top record and then **Shift**-click on the bottom record and all the records in between will also be highlighted.
- Holding the **CTRL** key down while clicking will extend the click to the entire record as if the record handle were clicked, thus selecting in accordance with selection modes. **CTRL** is used when tables are wider than the table window so that it is inconvenient to click on the record handle. We can then **CTRL**-click onto any cell to select that record and **CTRL**-**SHIFT**-click on a cell to select a range of records.
- To deselect a record, change the selection mode to **Subtract** and click on the record handle or **CTRL**-click on any cell in a record.

Records in tables may also be selected using the Query toolbar for rapid selection using simple query operators.

#### Showing Only the Selection in a Table

When working with large tables we may wish to pop open a table window that shows only records in the selection.

To show only the selection in a table:

1. Open the table.
2. In the main menu press in the **Selection Filter** button.
Showing a table with Selection Filter pressed in will dynamically update the table every time the selection changes either in the table or in an associated drawing. If we have a table with a thousand records in it and five are selected, then only five records will initially appear in the filtered table window and all of them will be highlighted. If we CTRL-click on one of them to de-select it (remember, CTRL-clicking on a highlighted record toggles the highlighting) it will disappear from that table.

See the View - Selection Filter topic for an example.

Query Toolbar Restricted to the Selection Filter

When the selection filter button is engaged subsequent operations of the query toolbar will apply only to those records that are shown in view by the selection filter. This makes it easy to run simple queries that refine the result of a previous query toolbar operation.

Suppose we select records with values in column A greater than 0. Suppose we now want to select records with values in column B greater than 0, regardless of what is in column A; that's easy to do as we simply run the query toolbar again.

Now suppose we again select records with values in column A greater than 0, but this time we also want to find within those selected records only those records that have values in B greater than 0 as well. After making the selection with the query toolbar to find those records with values in A greater than 0 we turn on the selection filter so that those records are shown and then next we run the query toolbar one more time to find those records that also have values in B greater than 0.

Selection using SQL

Structured Query Language (SQL) is an industry standard database query language used to select records from tables and to present those records with the fields desired. The result of an SQL query is what appears to be a new table. Although this is "selection" in the database sense of the word, the actual records and the objects with which they are associated are not selected in the Manifold sense of the word. They simply appear as their own table.

To use SQL to make a selection in the Manifold sense of the word we click open the SQL Query to see the table it creates and then we choose Edit - Select All to select all the records in that table.

See the introductory topics on SQL beginning with Queries. For reference information on SQL see the SQL Reference Guide information beginning with the SQL in Manifold System topic.

Saved Selections and Tables

Just like other components, we can open the selections pane and use it to save selections that we make in a table window. This is a very handy when making different selections and wishing to get back to a previously-made selection.

Selection Using More Like This

We can also select records in tables using the More Like commands. See the More Like This topic for more information.
**Formatting Columns**

We can change the appearance of values shown in a column by formatting the column. To do so, right click onto a column head and choose Format from the context menu. The same format will be applied throughout the entire column.

Format options will be displayed that are appropriate to the type of column. For example, a column for a floating-point field will have different **Style** options than a money field, integer field or text field.

**Style**

The overall type of format to be applied, for example, whether percentages are shown as .4565 or as 45.65%.

**Positives**

Enabled with numeric types. Controls the appearance of positive values.

**Negatives**

Enabled with numeric types. Controls the appearance of negative values. For example, negative currency values may be show.

**Separate thousands**

Enabled with numeric types. When checked, inserts a separation character to indicated thousands. The character used is determined by the current Windows locality settings, being a comma "," in the US and a period "." in various European countries.

**Alignment**

Left, Right and Center alignment. The Right alignment has the side effect of lining up decimal points when styles using a specified number of decimal places are used.

**Decimal places**

Enabled with numeric types. The number of digits to the right of the decimal point to display.

**Width**

The width of the column in characters.

**Indent**

The indent to apply in number of characters.

**Style Options**

Style options depend on the type of field. In general, style options are patterned on those found in Microsoft Office applications. Some field types have many style options while others have few options. Following are notes on the most popular options for field types that have many style options:

**Floating point**

<default> - Shows numbers without thousands separators and with as many decimal places are significant, truncating trailing zeros.

12.34 - Use the specified number of decimal places and other options.

12 1/3 (up to one digit) - Round to the nearest fraction using single digits in both the numerator and denominator. Examples: 6 5/7, 3 5/8 and 30 1/9.

12 17/50 (up to two digits) - Round to the nearest fraction using up to two digits in both the numerator and denominator. Examples: 5 21/50, 8 4/25, 3 3/4

12 1/2 (as halves), 12 1/4 (as quarters), 12 3/8 (as eighths), 12 5/16 (as sixteenths), 12 3/10 (as tenths), 12 34/100 (as hundredths) - Round to the nearest fraction using the specified denominator units.

1.23D+001 or 1.23E+001 - Scientific notation using "D" or "E" to indicate the exponential part.

1.23D+1 or 1.23E+1 - Scientific notation using "D" or "E" to indicate the exponential part with leading zeros trimmed from the exponent.

**Currency**

<default> - Same as floating point.
<no symbol> - Use the given number of decimal places without any currency symbol.

symbols - Use the given currency symbol to the given number of decimal places.

three letter codes - Use the given three letter SWIFT currency code.

Boolean <default> - Use 1 for true and 0 for false. Other options provide for the usual True / False, Yes / No and other renderings of Boolean status.

Date and Time <default> - American style month/day/year dates as in 3/09/2001. Note that date and time field types store not only the date but also the time in all cases. Whether the date, the time, or both together are visible depends on the formatting.

Latitude <default> - Standard decimal degrees where a minus sign indicates Southern latitudes and Western longitudes. Trailing zeros will be truncated.

12.3456 - Display using the given number of decimal places.

12°20.736 - Degrees with minutes and decimal fractions of a minute. A slightly dissonant format that nonetheless seems to be popular in certain GPS applications. It's used in some marine charts to note waypoints.

12°20'44.2" - Results in displays using degrees, minutes and decimal seconds formats.

N - The above styles are available in an N style, which appends an N, S, E or W to show North, South, East and West latitudes or longitudes. Otherwise, a minus sign will be used to indicate Southern latitudes and Western longitudes.

Note: Control the number of trailing digits after a decimal point with the Decimal Places option.

Percentage <default> - Show percentages using percent symbol, for example, 52.01 %

0.52 - Show percentages as decimal fractions, for example, 0.456.

options - Other options repeat the various fractional styles available in floating point numbers.

Editing

Cells that are formatted in a given column formatting style will accept edits in any compatible style. For example, if a date and time column is formatted to show dates as 3/09/2001 and one enters 9 mar during the year 2001 the entry will be accepted and shown as 3/09/2001. In this case, Manifold knows that "mar" in date and time fields means March. If the or time is not explicitly given the system uses the current system year and time. To specify all parts of the data and time value, enter explicitly those parts desired. If one enters 9 mar 99 the entry will be accepted and shown as 3/09/1999.

Latitudes and longitudes expect that a single number with a decimal point, like 90.1234 is intended as a decimal degrees value. Other numbers in the generic form of '90 2 3'-type strings will be parsed as degrees, minutes and sections using the two or three numbers with white space or almost any characters in between. This makes it easy to enter degrees, minutes and seconds values for latitude and longitude without hunting all over the keyboard for obscure ways of entering the "degree" character.

Other Uses of Column Formats

Column formats specified in tables will be used in labels and in export.
• Labels created from fields will use whatever formatting has been assigned to that field. See Creating Labels from Fields for examples.
• When tables are exported to purely text formats such as .csv or .html, the contents of the table fields will be formatted as specified by table formats.

Specifying Languages for Columns

Language specification in text columns varies depending on whether Unicode or ANSI text is used. Unicode text columns can simultaneously contain characters from more than one language. ANSI text columns are usually restricted to one language in addition to (possibly) English. Character data contained in an ANSI text column will be interpreted by the system according to the code page associated with that column.

By default, ANSI text columns use the system code page. If desired, any ANSI text column in a table can be set to use a different code page by right clicking on the column head and choosing Language.

Notes

Why the strange fractional styles for numbers? These are often used to express measurements using English units such as inches. In the USA, for example, people routinely use measurements such as 8 3/16 inches.

Tech Tips

From the context menu for table columns (right click onto a table column head) choose Best Fit, Best Fit All or Best Fit Titles to set width of table columns automatically.

Users are sometimes surprised to see numbers they expect to be values like 1.1 appear in a query or other table as 1.09999942779541, even if a floating point type is used. That is a result of the fundamental architecture of computers combined with the choice of format style used.

Floating-point values have finite precision. Some values, such as 1.2, or 0.25 can be represented precisely, but most can not. For example, neither 12.6 nor 13.7 can be represented precisely. To see what really hides behind a value shown as 12.6, set the format style of the relevant column to 12.34 and increase the number of decimal digits to 10. Try the same for 13.7.

If we subtract one approximation from another, as in the case of an SQL query, we get an approximation of the result so that an expected result such as 1.1 may appear as 1.09999942779541.

The default format style uses a Microsoft Windows API which makes some approximations, such as 12.6000..., look round but leaves others unaffected. There is some logic behind that API, but that logic is obviously not applicable to all cases. To provide fine control over desired appearance is one reason why Manifold allows setting the format style and the number of decimal digits explicitly.
Transfer Rules

This topic applies only to tables that are associated with a drawing. Transfer rules come into play when the transform toolbar for drawings is used to create new objects in a drawing or when commands such as Dissolve transfer fields between objects. A transfer rule may be set for each column (field) in the table that specifies how that field will be transferred to any new objects created by various transform operators. By default, all fields in a table are copied to the new record created for the new object.

To specify a transfer rule for a column:

1. Open the drawing's table.
2. Right click on the column and choose Transfer rules.
3. In the Transfer Rules dialog, specify the transfer method to be used for one to many (1 to N) operators. The Proportional methods available for numeric fields also require specification of a second numeric field.
4. Specify the transfer method to be used for many to one (N to 1) operators.
5. For text fields, choose any desired options to ignore case or whitespace.
6. Press OK.

Transfer rules may also be specified by opening a table using the Table - Design dialog and double-clicking into a transfer rule for a column.

The transfer rules specified for each field will be used whenever a transform operator creates a new object. There are two types of transform operators that create new objects:

- **One to Many (1 to N) Operators** - For each original object these operators create one or more new objects. An example is the Points operator, which creates a new point at each coordinate location that defines the shape of a line or area. Each line or area will therefore be used to create many new points. Operators like Centroids that create one new object for each original object are considered to be the simplest case of a one to many operator. The task for transfer rules for such operators is to specify how one value from the original object should be apportioned to the possibly many new objects created.

- **Many to One (N to 1) Operators** - These operators create a single new object from many original objects. The classic example is Convex Hull, which creates a single area object that covers a set of many points.

If a transfer rule is specified for both 1 to N and N to 1 for each column in a table Manifold will know how to transfer fields for objects created by any transform toolbar operator. By default, the Sample method is used for all fields. The Sample method is a reliable way of dealing with “many to one” (N to 1) operations so it is used by default.

Some Manifold users prefer to have the Copy method used by default for newly created fields. If the Copy method is desired, in Tools - Options - Miscellaneous check the box for the Set transfer rules for new columns to Copy / Copy.

**Note:** intrinsic fields, Active columns, rank columns and columns brought into a table via a relation do not participate in transfer rules since by definition they are computed for each object individually.

Transfer Methods

Transfer methods that may be specified in the Transfer Rules dialog include the following. Methods apply to all field types except as noted. Uncheck the Transfer column box to not transfer the field.

**One to Many (1 to N)**

- **Copy** Transfer a copy of this field's value to each new object created.
- **Equal Division** Divide equally the value by the number of new objects created and place equal shares into each new object. Numeric columns except latitude and longitude.
### Tables

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Proportional</strong></td>
<td>The field's value is distributed to the newly created objects in proportion to some other numeric column, for example <strong>Area (I)</strong> or <strong>Length (I)</strong>. This method is most often used with geographic fields such as <strong>Area (I)</strong> but could also be used to distribute values based on non-geographic numeric fields such as those that might code for budget or population.</td>
</tr>
</tbody>
</table>

**Many to One (N to 1)**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>Transfer the average value for this field in the origin objects into the new object. Numeric columns.</td>
</tr>
<tr>
<td>Copy</td>
<td>Pick one of the &quot;many&quot; objects at random and transfer a copy of this object's field's value to the new object created.</td>
</tr>
<tr>
<td>Count</td>
<td>Transfer a count of the number of objects used to create the new object. For example, if 10 points are used to create one convex hull area object the value 10 would be transferred. Numeric columns. Note: choosing <strong>Count</strong> will dominate the transfer even if it is a 1 to N transfer.</td>
</tr>
<tr>
<td>Maximum</td>
<td>Transfer the maximum value for this field in the origin objects into the new object. Numeric, date and text columns.</td>
</tr>
<tr>
<td>Minimum</td>
<td>Transfer the minimum value for this field in the origin objects into the new object. Numeric, date and text columns.</td>
</tr>
<tr>
<td>Median</td>
<td>Transfer the median value for this field in the origin objects into the new object. Numeric, date and text columns.</td>
</tr>
<tr>
<td>Sample</td>
<td>Arbitrarily choose a value from one of the originating objects and transfer it to the new object. The default for text columns.</td>
</tr>
<tr>
<td>Sum</td>
<td>Transfer the sum of values for this field in the origin objects into the new object. The default for numeric columns.</td>
</tr>
</tbody>
</table>

When working with text fields, the maximum, minimum and median values are chosen according to lexicographic ordering for the relevant text field. For ANSI text fields, lexicographic ordering depends on the code page. Using the maximum, minimum and median transfer rules with text fields is only recommended for experts wanting to exploit the effects of lexicographic ordering with a full understanding of how it works for the text fields at hand.

### Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Transfer column</strong></td>
<td>Checked by default. Uncheck to prevent this field from being transferred.</td>
</tr>
<tr>
<td>(1 to N field argument box)</td>
<td>Enabled for numeric fields when the Proportional method is selected. Specify the field, such as <strong>Area (I)</strong>, to be used to allocate the value.</td>
</tr>
<tr>
<td>Ignore case</td>
<td>Enabled for text fields. Treat upper and lower case the same when making lexicographic comparisons to determine the value to be transferred. If checked, &quot;San Juan&quot; would be considered the same as &quot;san juan&quot;.</td>
</tr>
<tr>
<td>Ignore leading and trailing whitespace</td>
<td>Enabled for text fields. Ignore any whitespace characters at the beginning or end of a text string when making lexicographic comparisons to determine the value to be transferred. If checked, &quot;San Juan&quot; would</td>
</tr>
</tbody>
</table>
be considered the same as "San Juan".

Ignore internal whitespace

Enabled for text fields. Ignore any whitespace characters that occur within a text string when making lexicographic comparisons to determine the value to be transferred. If checked, "San Juan" would be considered the same as "SanJuan".

Proportional

The Proportional methods transfer values using some other numeric field (usually the Area(I) or Length(I) intrinsic fields) as a guide. Active Columns may also be used to guide Proportional transfer. Suppose we will apportion the value 30 from one object into three objects that have guiding values of 2, 2 and 1. Using Proportional will transfer the value 12 into the first and second objects and 6 into the third object.

A typical use of Proportional is to allocate values when splitting up areas. Suppose we have a single country area that we wish to break into a group of provinces using the Split with transform and a set of boundary lines for the provinces. If we have a Population field for the country and we assume that population is evenly dispersed (not usually true in most countries, but we will assume so for this example) we would like each newly created province area to inherit a proportional share of the population of the country. To accomplish this, we would set the 1 to N transfer rule to be Proportional for the Population field and we would use Area (I) as the guiding field for proportionality.

When the country is split into smaller areas, the Area (I) for each will be computed and compared to the Area (I) of the original country. Each smaller areas will get a proportional share of the Population field from the country.

Example

In this example we use Convex Hull to create areas that cover a selection of points. We use transfer rules to sum the values of a field and place the sum into the newly created area.

We begin with a drawing that shows points. We have selected six points. We have also saved this selection to the selections pane so that we can call it up later.
The drawing's table shows that each object has a **Population** field. The records for the selected points are highlighted in red selection color in the table.

To specify the desired transfer rules, we right click onto the **Population** column head and choose **Transfer Rules** from the context menu. In the **Transfer Rules** dialog we choose **Sum** as the method for **N to 1** and press **OK**. This tells Manifold that whenever a transform operator creates one object from many, Manifold is to sum up the values of the **Population** field for the many objects and to place that summed value into the **Population** field for the resultant object.

The **Convex Hull** transform operator is an operator that creates one object from many. We can apply it to the [Selection] as seen in the transform toolbar above to create a convex hull for the selected points.

When we press **Apply** on the toolbar the convex hull area appears in the drawing.

A new record has appeared in the table as well. The new record has a **Population** value that is the sum of the **Population** values of the points that were used to create it.
We can see this is the case by clicking on the saved selection for the points used to create the convex hull and then pressing the Preview button in the selections pane. This highlights the saved selection in blue preview color. We can see that 275 is the sum of the population values highlighted in blue.

We can next select the other five points. We'll also save this selection in the selections pane so we can later preview the records for these points.

Seen in the table the records associated with these points are highlighted in red selection color.
If once more we apply the Convex Hull operator to the [Selection] in the transform toolbar we create a convex hull area about these five points.

In the table window we can see that another object has been added, with a Population value of 201.

If we click the saved selection for the second set of points and press the Preview button in the selections pane we can see that 201 is the sum of the Population values for the points (highlighted in blue preview color) used to create the convex hull area.

For Advanced Users

The "1 to N" and "N to 1" categories of transfer methods may make it seem that either one or the other method is used in any given situation. In actuality, both methods are used all of the time in all cases. First the N to 1 method is applied to the source set and then the 1 to N method is applied to the destination set of values. Careful contemplation of the operators involved will show that results will always be the same as if only one or the other method were used, except in the case of the Count operator in "1 to 1" operations.
The simultaneous operation of \(1 \text{ to } N\) as well as \(N \text{ to } 1\) can be seen by considering a case of \(1 \text{ to } 1\) transfer. Suppose we make a copy of a drawing of areas. The original we call drawing \(A\) and the copy we call drawing \(B\). Suppose we delete all data attribute values from \(B\). We make changes to values in \(A\) and then we want to transfer the new values to \(B\). Because \(B\) was originally a copy of \(A\) we know that the areas in \(B\) are exactly coincident with the areas in \(A\). This is a "1 to 1" transfer since the value of each area in \(A\) will be transferred to one, and only one, area in \(B\).

Our question is which transfer methods operate... the \(1 \text{ to } N\) or the \(N \text{ to } 1\)? The correct answer is "both" as we can see from the following examples, each of which consider what happens to a value from a single area \(A\) with a numeric value of 5 that is transferred to the corresponding area \(B\):

Let's say our \(1 \text{ to } N\) rule is "copy" and the \(N \text{ to } 1\) rule is "maximum". When transferring values from \(A\) to \(B\) the system will first apply the \(N \text{ to } 1\) rule to the source set of values (consisting of a single value from area \(A\)) yielding a "maximum" of 5, and then second apply the \(1 \text{ to } N\) rule with a target set of values (consisting of the single value of area \(B\)) "copying" 5 from the previous step to each value. The value of \(B\) will be set to 5.

Let's now say our \(1 \text{ to } N\) rule is "equal division" and the \(N \text{ to } 1\) rule is "sum". When transferring values from \(A\) to \(B\) the system will first apply the \(N \text{ to } 1\) rule to the source set of values (consisting of a single value from area \(A\)) yielding a "sum" of 5, and then second apply the \(1 \text{ to } N\) rule with a target set of values (consisting of the single value of area \(B\)) distributing 5 from the previous step "equally" to each value. The value of \(B\) will be set to 5.

Let's now say our \(1 \text{ to } N\) rule is "sample" and the \(N \text{ to } 1\) rule is "count". When transferring values from \(A\) to \(B\) the system will first apply "count" to \(A\) to yield 1, and then second will "sample" 1 to \(B\). The value of \(B\) will be set to 1.

Summarizing the above, a "1 to 1" transfer will almost always result in copying the value from source object to target object, unless the \(N \text{ to } 1\) rule is set to "count", in which case the result will be 1.

Note that in cases that are truly many to one the resultant value always appear as if only the \(N \text{ to } 1\) rule is consulted. That is because Copy, Equal Division, Sample and Proportional all have the same effect when applied to the target set of values if the target set only has one member. Likewise, in cases that are truly one to many the resultant value always appears as if only the \(1 \text{ to } N\) rule were consulted. That is because all of the possible \(N \text{ to } 1\) operators have the same effect when there is only one member in the source set.

A fine point: although both \(N \text{ to } 1\) and \(1 \text{ to } N\) rules should always be understood to be in force Manifold will optimize execution so that no \(N \text{ to } 1\) computations are performed if it is clear that only one item would be in the "N" set.

**Tech Tips**

Have a lot of columns in your table and want to see all the transfer rules at once? Open the table and then choose Table - Design to see a summary.

Transferring column values between objects in a linked drawing using Sample or Copy N-to-1 aggregates will choose blank strings over NULLs.

**See Also**

Selection for examples of selections and using the selections pane to save and preview selections.

Transform Toolbar - Drawings and Transform Operators - Drawings for information on using the transform toolbar in drawings.

Spatial Overlay - Spatial overlays are a different concept than the idea of using transfer rules to aggregate or to allot values when creating objects with transforms. However, it is a related concept in that aggregations or allotments are used to transfer field values between objects.

Transfer Contour Line Height to Points - A simple example using Spatial Overlay and transfer rules.

Color Areas by Counts - A simple example using Spatial Overlay and transfer rules to color areas by the number of points they contain.
Shortest Path over Land - A complex example that uses Spatial Overlay and transfer rules.

Intersection Overlays - A small, but sophisticated, example that uses Transfer Rules, Split with, Clip with (Intersect) and Spatial Overlay commands with areas.
Relations

Relations are connections between two tables in a Manifold project that allow one table to show columns from another table. A relation between tables uses a key field with unique values common to both tables to connect the two tables. The same field type must be used in the columns being related. This allows data that appears in one table to also appear in another table without having to physically duplicate the data.

To form a relation between two tables

1. Open the table that is to display the additional fields and click on Table - Relations.
2. Click on the New Relation button. This launches the Add Relation dialog.
3. In the Add Relation dialog, choose another table from the list box.
4. In the Add Relation dialog, click on one field for each table that will be used to match records and press OK.
5. Back in the Relations dialog, check the desired columns from the other table. Press OK.

Note: The New Relation button will not be enabled unless there is at least one other table in the project. To form a relation with a table using an external data source, first link a table in the project from that external data source. This creates a new, linked table in the project. We can then form a relation with that new table.

Columns that are "borrowed" from another table will appear in the table with yellow background color to indicate they are imported. They may be used like any other column, for example, for sorting, filtering, within formulas, within rank columns or for thematic formatting. Tables may have more than one relation with more than one other table.

Columns that are included from other tables may not be used as a key field to form a new relation. Columns can only be linked through one relation. They cannot be passed on in turn through yet another relation. For example, if table A has a relation with table B so that columns B1 and B2 appear in table A, table C cannot form a relation with table A and "borrow" columns B1 and B2 from table A. To include columns B1 and B2 in table C, table C must form a relation directly with table B.

Relations Dialog Commands

- New Relation - Add a new relation.
- Delete - Delete highlighted relation.
- Include All - Show all columns from the related table in this table. Check all checkboxes.
- Include None - Do not show any columns from the related table in this table. Uncheck all checkboxes.
- Include Inverse - Show all columns from the related table that were hidden and vice versa. Toggle all checkboxes.
- Properties - Edit the highlighted relation by calling the Add Relation dialog.

Example

Relations are used to add columns to tables from other tables.
Suppose we have a Customers table with a CustomerID field and a ContactName field.

Suppose also we have an Orders table with OrderID, CustomerID and an OrderDate field.

We would like to create a relation that shows in the Customers table the OrderID and OrderDate columns so that for each customer we can see all orders for that customer as well as the date of order. To do this, we open the Customers table and choose Table - Relations.

In the Relations dialog we click the New Relation button to add a relation. This opens the Add Relation dialog.

In this dialog we choose the Orders table in the list box and then highlight CustomerID as the linking field in both the Customers table (the left pane) as well as in the Orders table (the right pane, under the list box showing the Orders table). Note that we could have used any two fields that contain matching values even if they are named differently. However, as a matter of sensible database organization it makes sense to give the same name to the field used to save customer identification in all tables. We press OK.
The result back in the Relations pane is that we have a new relation listed as CustomerID: Orders.CustomerID. This is a shorthand way of saying that this relation is determined by matching values in the CustomerID field in this table to the CustomerID field in the Orders table.

We check the fields (columns) we would like to include from the Orders table and press OK.

The result in the Customers table window is that two new columns appear, the contents of which are taken from the Orders table. These contents are shown in yellow, which is a generic background color used to indicate values that are computed or otherwise derived.

Converting to Regular Columns

Right click on a relation’s column head and choose Flatten to convert the column to a regular data column. The current value of each cell will be used to populate the regular column and the link to the other table will no longer exist.

Notes

A table may have more than one relation. For example Table A might include two columns, B1 and B2, by way of a relation with Table B and Table A might also simultaneously include one more column, C1, by way of a relation with Table C.

Relations can exist between any tables that exist in the project. In particular, a drawing’s table can have a relation with another table.
Relations can exist between tables that are imported into a project and tables that are linked into a project. If we only have one table in a project we cannot form any relations because there is no other table with which to form a relation.

When including fields from tables that are linked into a project, keep in mind that tables linked into a project may be provided by files or OLE DB providers that might be participating in multi-user sessions with other programs. That's usually the objective of including such external tables, of course, but one should not be surprised if data in columns that are included from external tables appears to change without user intervention.
Attaching External Tables to Drawings

External tables may be attached to drawings by using relations. Relations allow us to "borrow" columns from the external table and to show them and use them as if they were a part of the drawing's own table. In this topic, an external table is meant as any table other than the drawing's own table. It can be a native table entirely within the project, or it can be a linked table that might possibly get its contents from some other data source.

To attach an external table using a relation, we need to have at least one field in the drawing's table that can be used to form the relation with the external table. Once the relation is formed we can include fields from the external table within the drawing's table.

To Attach an External Table to a Drawing

1. Link the external table into the project using File - Link - Table
2. Examine the drawing's table and the external table to find a field in each with values in common.
3. Open the drawing's table and use Table - Relations to form a relation with the external table using the fields in each that have values in common. Include in the drawing's table those fields from the external table that are desired.

Note that the drawing's table must have at least one field that can be used as a key field to form a relation with the external table.

Example

We have a drawing called Mexico with an associated table called MexicoTable. The drawing shows provinces in Mexico as areas.

In addition, we have used File - Link - Table to link an external table, called Sales, into the project as seen in the project pane shown above. The Sales table originates in an Access database that is part of our sales management system. Note that the Sales table is shown with the "table and cylinder" icon that is used to denote external tables that are linked into a project.

If we open MexicoTable, we see it has only two fields: the object ID field connecting the records in the table to the drawing's objects and a Place_name field that gives the name of each province.
If we open the **Sales** table, we see it has two fields: a **Sales** field giving sales numbers for each province and a **Province** field giving the name of each province.

![Add Relation Dialog](image)

If we click on the **MexicoTable** table to move the focus onto that window and choose **Tables - Relations** we can create a relation between that table and the **Sales** table. Shown above is the **Add Relation** dialog in this procedure, where we have clicked on the **Place_name** field in the **MexicoTable** pane and on the **Province** field in the **Sales** pane to indicate that values in these two fields should be matched to form the relation. In the **Relations** dialog we check the box for the **Sales** field in the **Sales** table to show it within the **MexicoTable** table.

![MexicoTable](image)

The result is that the drawing's table becomes equipped with a new column, called **Sales**, which is borrowed from the external **Sales** table using a relation. This **Sales** column behaves exactly as though it was a native part of the drawing table, however, it originates in the external table outside of Manifold. If our sales management system changes the Access database from which the **Sales** table is maintained, the values in the drawing's table will change as well.

**Notes**

The example uses simple tables with only one relation in the drawing's table. We can use more than one external table if desired by specifying more than one relation with external tables.

What do we do if we would like to use an external table with a drawing but there are no fields in the drawing? For example, suppose we have downloaded a drawing of Mexico with areas but there is no "place_name" or equivalent field in the drawing? In that case, we will have to manually add a field to the drawing's table and manually add a unique identifier to each record that can be used to match records with the external table we would like to use.
A nuance: We have used a simple drawing that has only one area object for each province. As a result, there is only one record for each province. The sample Mexico drawing imported from the BTS North American Transportation Atlas Database provided with Manifold System is not such a simple drawing. Several of the provinces in that drawing are made up of two or three areas to represent major islands that are part of the provinces. Such provinces have two or three records for each province in their tables.

See Also

The Database Console provides a handy way of browsing external databases to see what tables they contain and the names of and types of fields in the tables.

Drawings and Databases
Drawings and Tables
Geometry in Tables

This topic discusses the storage of geometry within tables, whereby binary data representing the shapes and locations of objects is stored within the table. That is a more advanced usage than the relatively simple case of geocoded tables, where some columns give lists of X/Y or latitude and longitude locations. For the more simple case of geocoded tables, see the Linked Drawings from Geocoded Tables topic.

Let's begin with a refresher on nomenclature: the objects in a drawing are the areas, lines and points that make up that drawing. A drawing's table has a record, a horizontal row, for each object. If there are any data attributes attached to the drawing's objects, those attributes will appear as columns in the table. The information that defines the locations of points and the locations and shapes of lines and areas is called the geometry of the drawing. The metric of a specific object is the geometric data that defines that object.

When working with drawings in Manifold we quickly become used to the idea that there are two parts to a drawing: the drawing component itself and the drawing's table.

When we pop open the drawing component in a drawing window it paints the objects in the drawing for us.

![Drawing Component]

Because GIS as done interactively is primarily a visual activity, drawings and other visual displays take the lead role in Manifold’s user interface.

When we pop open the drawing's table in a table window we see any data attributes in a classic row and column tabular presentation. Data attributes in tables are normally the sort of information that is easiest to comprehend and edit within a classic, text, row and column tabular user interface.

The way Manifold is taught to beginners together with the routine use of drawings and the tables of drawings conveys the idea that the geometry of objects is stored within the drawing and the data attributes are stored in the
drawing's table, as if they were two different storage locations. But the truth is deeper, more general and more powerful than that.

The truth of the matter is that inside Manifold all information for drawings, both geometry and attributes, is stored within tables. It is just a question of how different types of information are best displayed and used. Since geometry is best understood using a visual interface, by default the geometry in a table is shown using a drawing. But the geometry is still in the table.

We can see the geometry data within every drawing's table by turning on the Geom (I) implicit field. The Geom (I) column reveals the geometry column of a table, wherein the metric of each object is stored using Manifold's own Geom data type, a form of binary data that encodes an object's metric.

![Drawing Table](image)

<table>
<thead>
<tr>
<th>ID</th>
<th>Color</th>
<th>Geom (I)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Red</td>
<td>&lt;geom, area, branches: 1, coords: 3&gt;</td>
</tr>
<tr>
<td>2</td>
<td>Blue</td>
<td>&lt;geom, area, branches: 1, coords: 6&gt;</td>
</tr>
<tr>
<td>3</td>
<td>Green</td>
<td>&lt;geom, area, branches: 1, coords: 32&gt;</td>
</tr>
<tr>
<td>4</td>
<td>Yellow</td>
<td>&lt;geom, area, branches: 1, coords: 4&gt;</td>
</tr>
</tbody>
</table>

It is impractical in a text, tabular presentation to convey a visual impression of geometry so instead when the Geom (I) column is turned on a table will report some summary information about the metric for that object: the type of geometry data, the type of object (area, line or point), the number of branches and the number of coordinates.

Since geometry is best comprehended visually, Manifold uses a visual, drawing user interface to present geometry. But the geometry that is displayed or edited in a drawing window is still ultimately stored within the table.

When we edit the shape of an object in a drawing, we are simply taking advantage of the drawing as a handy, visual user interface to edit the contents of the Geom (I) field for that object in the table. For example, we can select an object for editing in a drawing.
The object's row is also selected in the table. We can use that to follow along in this example.

Deleting a coordinate in the object changes both the geometry and the visual appearance of that object.

We can see by the coords: 6 value changing to coords: 5 that the geometry in the table has also been changed (there is now one less coordinate in the metric). The drawing is simply a visual interface to geometry stored in the table.

When we create a drawing the two part user interface system for visualizing drawings and drawing tables is automatically put in play by Manifold.

Manifold automatically creates a drawing component with the drawing's table indented under it in the project hierarchy to indicate the association between the two. But this is just a choice of user interface to present two icons, a drawing icon and a table icon, upon which users can click depending on whether they want to see the
data as a visual drawing or as a table. A different choice in user interface design might have simply used one icon and then given users the choice to click it open as a drawing or to click it open as a table.

In fact, within Manifold we can create tables that contain geometry which do not automatically have a drawing window user interface associated with them.

In the above example, we can copy the **Drawing Table** component and paste it as a new table called **Drawing Table 2**. This is now simply a table not associated with any drawing.

![Drawing Table 2](image)

After simplifying the illustration by hiding undesired columns and renaming the geometry column (to emphasize for the purposes of this illustration that it is not an intrinsic column anymore), we get the display seen above. This is now simply a table that contains a geometry column. We can link a drawing to it, write a query that grabs a subset of data from it, export it to an external DBMS or do many other things with it.

**Exploiting Geometry in Tables**

Once we understand that geometry is stored in tables we can exploit that understanding in many useful ways. We can:

- Store tables within external databases, including the geometry of objects.
- Translate geometry within tables from Manifold format to other formats for geometry.
- Export tables containing geometry to external databases, in effect a way of storing drawings within external database systems.
- Import tables containing geometry from external databases, in effect a way of importing drawings stored in databases.
- Write queries that dynamically select desired geometry from tables.
- Link drawings from queries that select geometry.
- Link drawings from tables in external databases, in effect a way of letting a DBMS system maintain a drawing that potentially thousands of simultaneous Manifold users might be editing simultaneously.

If an external DBMS can store data in binary form (as can all of the big-name DBMS packages), we can store drawings within external DBMS servers by simply exporting tables containing geometry to those servers.

Note that the realization that the geometry of a drawing is encapsulated within the drawing's table as a **Geom (I)** column is different than the simple notion of a geocoded table. A geocoded table is a way of maintaining a list of point locations in a table. It is simply a table that for each record contains latitude and longitude columns specifying where that record is located.

Geocoded tables are useful mainly for points and cannot store areas at all and lines to only a limited degree. Geocoded tables are a low performance, limited solution. Drawings linked from geocoded tables are read-only and cannot be edited. The main reason Manifold provides a simplified way of linking drawings to geocoded tables
is that lists of latitude / longitude locations in tables are extremely common and it is likewise a common desire to see such locations displayed in a map. So Manifold provides a simple way of doing that.

In contrast, tables that contain a geometry column have the ability to fully specify drawing geometry with all the characteristics sophisticated GIS usage requires of objects: that they can be areas, lines or points; that they can be multi-branched for topological sophistication, that they can exist within a specific coordinate system. To distinguish tables containing a geometry column from simple geocoded tables we often refer to them as **spatial tables** within Manifold documentation.

When stored within a fast DBMS spatial tables can be a very high performance solution that scales well for use by the very largest enterprise applications. Spatial tables provide tremendous flexibility and power. Drawings linked from spatial tables are editable, assuming the connection technology used is read/write and the user also has write permissions for the DBMS table used.

The usual way of storing spatial tables is within a spatial DBMS. Manifold makes it easy to store geometry data within spatial DBMS.

### Using Geometry Data in Tables

Why store geometry within spatial tables? Doing so allows use of linked drawings that are dynamically created from tables or queries containing geometric data. Because of the great flexibility of SQL the ability to create linked drawings from tables or queries makes it possible to create drawings on demand to suite a wide variety of purposes that cannot be fulfilled by static drawings.

Besides the obvious benefit of creating linked drawings on the fly as desired, there are many practical gains from storing geometry data in tables. To name just a few examples:

- We can store metric data in geometry columns in order to centralize all spatial data within a database, perhaps a central DBMS using SQL Server.
- We can store metric data in geometry columns to have more than one metric for a drawing object. For example, we could associate one object with another object without using any intermediate columns such as IDs, thus avoiding any risks of broken associations. Or perhaps we might want to store different versions of an object’s metric for different projections or for different users.
- We can use geometry values to store temporary results of geometric computations within queries.
- We can edit the **Geom (I) intrinsic field** in tables associated with drawings to edit the objects in the drawing.
- We can transform geometry data into different types to enable interoperability with different GIS systems.
- We can transform geometry data into native geometry types provided by spatial DBMS products like Oracle Spatial, to take advantage of server-side facilities provided by such products.
- Drawings linked from geometry in tables can be edited.
- If we use Enterprise Edition, drawings linked from geometry in tables can be edited by multiple users at the same time.
- If we use Enterprise Edition and a spatial index within a spatial DBMS, drawings linked from a spatial table can be restricted to a given area of interest. This enables desktop Manifold installations to be used to work with a manageable subset of what could be an immensely large data set.

### Geometry Types in Tables

Tables can contain geometric data columns that may use the following data types:

- Oracle **SDO_GEOMETRY** data, within Oracle databases that support this data type. This data type occurs only within Oracle tables and is dynamically mapped on the fly back and forth between Manifold's own **Geometry** type within tables in Manifold projects during OCI connections.
- IBM **ST_GEOMETRY** data, within IBM DB2 databases that support this data type. This data type occurs only within IBM tables and is dynamically mapped on the fly back and forth between Manifold’s own **Geometry** type within tables in Manifold projects during native DB2 interface connections.
- The **Geometry** type (also called **Geom**) is specific to Manifold System and contains the object type, the object metric and the coordinate system. **Geom** data can be stored within Manifold projects or saved into almost any DBMS system that supports binary “blob” storage, such as Access or SQL Server or MySQL. Exchanging data with Oracle data sources via the Oracle Call Interface (OCI) automatically maps Manifold **Geometry** values into Oracle **SDO_GEOMETRY** values (collections of geometric
Geometry, Geometry (SDE), Geometry (SHP) and GeometryWKB data types are referred to as Geom, GeomSDE, GeomSHP and GeomWKB as SQL data types and in coding. Therefore, these shorter names for these geometry data types are the usual way experienced Manifold users refer to these data types. For example, a Manifold user might say “I need to convert that Geom to a GeomSDE here.” In addition, the GeomWKB type is often abbreviated even more to simply WKB. This documentation will use either the longer forms or, more frequently, the shorter names for these data types.

Geom, GeomSDE, GeomSHP and GeomWKB data types can be implemented within any DBMS that can store binary data. In addition to tables within Manifold System, many modern database systems, such as SQL Server, Oracle and Access can store data in this form. Manifold can work with external tables in an external DBMS provider and save in Geom, GeomSDE, GeomSHP or GeomWKB form.

Geom and GeomWKB can be used within SQL Server using the Manifold Spatial Extender for SQL Server. This is by far the highest performance for storing drawings with a spatial index in SQL Server, especially when Geom is used. See the Spatial DBMS topic for information on the spatial extender.

GeomSHP and GeomSDE are somewhat similar approaches with differences in binary data organization. GeomSHP stores coordinates as double-precision floating point values. GeomSDE stores coordinates as integer values which are compressed using a simple run-length encoding scheme. The coordinate values must be scaled and shifted using the information stored in metadata tables.

Because GeomSDE depends upon metadata tables for meaning, some assumptions must be made when converting other geometry types into GeomSDE or when importing or linking a drawing from a table using GeomSDE. In both cases the integer coordinate values within GeomSDE values are assumed to use offsets of 0 and a scale of 1.

The Geom data type captures coordinate system information, but the GeomSDE, GeomSHP and GeomWKB types don’t. When used with projection-aware algorithms, values of the GeomSDE, GeomSHP and GeomWKB types are assumed to be in Latitude / Longitude. Working with values of these types in a projection other than Latitude / Longitude requires assigning these values the correct projection, for example, by converting to the Geom type and then using the AssignCoordSys query function.

Oracle SDO_GEOMETRY is available only within Oracle DBMS products. Manifold Geom data and Oracle SDO_GEOMETRY data are seamlessly mapped into each other during exchange whenever OCI is used to connect to Oracle databases. To force use of generic, binary-style storage instead of the native Oracle SDO_GEOMETRY storage within Oracle, connect to the Oracle data source using OLE DB or ODBC or (read only) ADO .NET.

Similarly, IBM ST_GEOMETRY is available only within IBM DBMS products. Manifold Geom data and IBM ST_GEOMETRY data are seamlessly mapped into each other during exchange whenever a native DB2 interface is used to connect to IBM DB2 databases. To force use of generic, binary-style storage instead of the native IBM ST_GEOMETRY storage within DB2, connect to the DB2 data source using OLE DB or ODBC or (read only) ADO .NET.

A simple rule for choosing from the above types is to always use a spatial DBMS vendor's native type. Use SDO_GEOMETRY with Oracle and ST_GEOMETRY with DB2, for example. Otherwise, use Manifold’s Geom type. In rare cases where interoperability with open source foolishness is more important than performance or reliability, we might be forced into using WKB. It almost never makes sense to use the ESRI types unless for the very limited case of interoperability with data held hostage in ESRI form.
Choosing a Geometry Data Type for Interoperability

One reason to store geometry in tables within database systems is to allow other applications to use such data. If we don't care about legacy applications and simply want our spatial DBMS to run as fast as possible while enabling multi-user concurrent editing we can simply use Manifold Geom type and either SQL Server with the Manifold spatial extender or Oracle (with Geom being automatically mapped into Oracle's SDO_GEOMETRY for us) or IBM DB2 (with Geom being automatically mapped into IBM's ST_GEOMETRY). If we want interoperability with other applications we have to decide what DBMS to use and also what geometry type to use.

When database tables containing columns of the above geometry types are saved from Manifold to an external DBMS provider, Geom values will be mapped to SDO_GEOMETRY if native Oracle interfaces are used or ST_GEOMETRY if native DB2 interfaces are used. In other cases, the data in such columns will appear as generic binary data

If the geometry data is in GeomWKB form, any application that understands GeomWKB will be able to use it. Although there are some, mostly academic, applications that can read or write GeomWKB, very few commercial applications have been deployed that use it (for reasons set forth below).

If the data is in Oracle SDO_GEOMETRY data then any application that understands SDO_GEOMETRY will be able to use it. Using Manifold + Oracle therefore will usually be the default choice for many people since more commercial applications can directly read or write SDO_GEOMETRY geospatial data than any other format.

IBM DB2 spatial users will want to use ST_GEOMETRY to allow interchange with any application that understands ST_GEOMETRY. DB2 also has a strong following, albeit with fewer applications than Oracle at this writing that can exchange spatial data with existing data sources managed by Oracle at this writing that can exchange spatial data using IBM's ST_GEOMETRY data type.

At first glance GeomSDE or GeomSHP might appear to be a way to allow easy interoperability with ESRI products, but what are commonly understood as ESRI "geodatabase," "ArcSDE" or "personal SDE" formats are not at present well supported by GIS industry products (not even by all legacy ESRI products) nor has ESRI encouraged interchange by other vendors using these data types. They remain closed, largely undocumented formats that have not been widely adopted outside ESRI. They also suffer from dependencies upon external metadata which makes them harder to use than self-contained types. The primary use for these types in Manifold is to facilitate data exchange with existing data sources managed by ESRI products. Their use in other cases makes little sense.

Choosing Between Geom and GeomWKB Types

Suppose we are working with a DBMS like SQL Server or MySQL that requires us to choose between GeomWKB and Manifold's own Geom type. Given that other applications are emerging that also understand GeomWKB, why would Manifold have a Manifold Geom type for storing geometry in tables? Would it not be easier to always use GeomWKB?

The reason Manifold has its own Geom type in addition to GeomWKB is that GeomWKB is limited. It does not contain the coordinate system in use, nor does it provide the ability to embed a variety of useful geometric information such as location precision or other data that can be cached to improve performance.

A good plan, therefore, is to use GeomWKB if one must have interoperability with existing applications that support it and use Geom for all other things. The storage impact for coordinate system and other data that are stored in Geom is not very large (typically less than 40 bytes and sometimes as little as 2 bytes per record) so using GeomWKB instead of Geom does not provide significant space savings.

On the other hand, using Geom lets us transfer tables with geometric data between machines without having to worry about coordinate systems. It also lets us have a table with objects that are in different coordinate systems, which will be automatically re-projected to the coordinate system of the linked drawing when it fetches data from the table.

Geom is also the native geometry type of Manifold itself. This has the utility of being always available in any drawing within Manifold without need of conversion to a generic GeomWKB type.

GeomWKB (WKB) and WKT

WKT is the "well-known text" format used by OpenGIS and similar standards. It is a human-readable text analog of GeomWKB. When viewing WKB data in tables, the data can be reported either in &lt;geom wkb, type&gt;
condensed format (the default), which only shows the type, or in WKT format, which shows both the type and the coordinates. To switch between the formats, right-click the WKB column, and select Format.

We can also transform GeomWKB data into WKT with CStr or CAST to create a text column containing WKT:

```
SELECT CStr(CGeomWKB(Geom(ID))) FROM [Drawing];
```

Queries and Geometry

Queries are used to create and alter columns containing geometric data. The SQL name of the Geometry type is Geom. The SQL name of the Geometry (SDE) type is GeomSDE. The SQL name of the Geometry (SHP) type is GeomSHP. The SQL name of the Geometry (WKB) type is GeomWKB. Using queries we can create geometric data from existing objects or literal values, we can convert geometric data between various representations and we can transform geometric data within tables. We can also use queries to create subsets of geometric data or to combine data from different tables. See the Spatial Extensions topic for a list of SQL functions that operate on Geom, GeomSDE, GeomSHP or GeomWKB data.

Creating a Table with Geometry

Tables storing geometry data are created from drawings with queries using the Geom (I) intrinsic column or the Geom(ID) call (which takes the value of the ID column and extracts the Geom for that value using the Geom(I) function).

Creating a Table storing a Drawing’s Geometry

1. Create or import the drawing into the project.
2. Write a query extracting the geometry data from the drawing.
3. Run the query.
4. Save the query results as a table.

A typical query that captures a drawing’s geometry as the Geometry column type as well as all data fields in the drawing would be...

```
SELECT Geom(ID) AS [Geom], [Drawing].* FROM [Drawing];
```

...where Drawing is the name of the drawing.

The query can then be used to create a table. To capture a drawing's geometry as the GeomWKB column type we first use Geom(ID) as above and then use the CGeomWKB(geom) function to convert the data to WKB form. These two functions may be nested, as in

```
SELECT CGeomWKB(Geom(ID)) AS [GeomWKB], [name of drawing].* FROM [name of drawing];
```

To covert geometry in Geometry or GeomSHP form into GeomWKB form, use CGeomWKB. To covert geometry in Geometry or GeomWKB form into GeomSHP form, use CGeomSHP. For example, after creating a drawing containing a GeomWKB column using the query above we could create another drawing containing the same data within a Geometry column using

```
SELECT CGeom(GeomWKB) AS [Geom] from [name of geom drawing];
```

An alternate way of creating a table that stores a drawing’s geometry is to use a Select Into query. This is used when creating a table in the same project. Create and run a query like the following:

```
SELECT Geom(ID) AS [Geom], [Drawing].* INTO [Table] FROM [Drawing];
```
where **Drawing** is the name of the drawing.

**Example**

In this advanced example we will create a table containing geometry from the **Mexico_eg** example drawing found on the Manifold CD. We will then use that table to create a linked drawing.

This is a very artificial example in that normally the use of spatial data is with a spatial DBMS. When Manifold exports a drawing to the spatial DBMS there is no need to explicitly create a geometry column within a drawing and to literally and explicitly export that geometry column. Manifold will simply export geometry to the spatial DBMS as needed when exporting a drawing. This example takes a different approach in that it exports geometry in a highly manual way, as if it were a binary field, without taking advantage of the automatic export of geometry for drawings to a spatial DBMS as is normally done.

However, for advanced users there is useful education to be had in explicitly creating geometry and then exporting it to a table. This exposes the inner workings of geometry and shows how it plays a role in defining objects in drawings.

![Mexico_eg Drawing](image)

![Mexico_eg Table](image)

We import the **Mexico_eg** file into our project to create a drawing and the drawing’s table.

![Mexico_eg Drawing](image)

The **Mexico_eg** drawing shows the provinces of Mexico. It is in latitude / longitude projection.

<table>
<thead>
<tr>
<th>ID</th>
<th>SQMI</th>
<th>SQKM</th>
<th>POBL...</th>
<th>VIVIEN...</th>
<th>PE.ACT</th>
<th>PEA_OG</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>29635.63</td>
<td>63068</td>
<td>2003167</td>
<td>381821</td>
<td>541908</td>
<td>529016</td>
</tr>
<tr>
<td>2</td>
<td>32189.13</td>
<td>73252</td>
<td>1276323</td>
<td>239541</td>
<td>306734</td>
<td>294498</td>
</tr>
<tr>
<td>3</td>
<td>52374.05</td>
<td>123181</td>
<td>1349378</td>
<td>263009</td>
<td>359994</td>
<td>347275</td>
</tr>
<tr>
<td>4</td>
<td>75106.63</td>
<td>182050</td>
<td>1823606</td>
<td>382914</td>
<td>577205</td>
<td>562366</td>
</tr>
<tr>
<td>5</td>
<td>23803.34</td>
<td>58329</td>
<td>2294954</td>
<td>425755</td>
<td>674431</td>
<td>669965</td>
</tr>
<tr>
<td>6</td>
<td>31400.15</td>
<td>79384</td>
<td>2249581</td>
<td>494118</td>
<td>710347</td>
<td>684550</td>
</tr>
<tr>
<td>7</td>
<td>25202.05</td>
<td>64924</td>
<td>3090726</td>
<td>674157</td>
<td>1036770</td>
<td>1009504</td>
</tr>
<tr>
<td>8</td>
<td>29614.38</td>
<td>69921</td>
<td>1665855</td>
<td>373476</td>
<td>576395</td>
<td>565471</td>
</tr>
<tr>
<td>9</td>
<td>29732.4</td>
<td>73475</td>
<td>317754</td>
<td>68479</td>
<td>104980</td>
<td>102763</td>
</tr>
</tbody>
</table>

If we open the drawing’s table we see that it has various fields for each province.

We will use this drawing to create a table that captures both the drawing’s geometry and all data fields in the drawing within the table.
To do so, we create a query with the above text and run the query.

```
SELECT Geo.ID AS [Geom], [Mexico_eg Drawing].* FROM [Mexico_eg Drawing];
```

The query captures all of the drawing's fields and includes a new geometry field. Manifold reports statistics for `Geom` type fields including the type of object, how many branches it contains and the number of coordinates that define it.

The query also captures the object `ID` field from the drawing because the query text we used simply selected all of the fields from the drawing. We don't need to capture this field since it will not necessary to capture the geometry of each object. We are including it simply because it is easier to write a query that selects all columns than it is to write a query that selects all of the columns instead of the `ID` column.

We can save this query to a database table. In this example we will save it to an Access `.mdb` file in a temporary folder. To do so, with the focus on the query results window we choose `File - Export - Table`.

In the `Export` dialog we choose `MDB Files (*.mdb)` in the `Save as type` box and provide a name for the file.
In the Export MDB File dialog we choose Access 2000 as the MDB Subtype and press OK. [Moments after pressing OK we realize that we could have unchecked the ID box to not export that field, but that's OK - we will drag it along even though we don't need it.]

At this point we have created an Access .mdb file called Mexico_eg Geometry in our tmp folder. The .mdb file contains one table that contains a record for each object in the Mexico drawing. Each record has a column for geometry data defining the area object and columns for the various other data attributes. In an Access .mdb file, the geometry data is stored as binary data. Manifold can interpret that geometry data to create the line, point or area objects within a linked drawing.

Let's now pretend that we've not just created this table, but rather that it was sent to us by a colleague. We know the table contains geometry data that can be used to create a Manifold drawing, so we will link it into our project and use it to create a linked drawing.

We begin by linking the table from the external .mdb file into our project. To do so, we choose File - Link - Table.
In the Link dialog we choose MDB Files (*.mdb) in the Files of type box and then we navigate to the .mdb file and double-click on it.

Manifold will pop up a subsidiary dialog asking what table we want to import, and we choose the only table in the .mdb file. The default name of the table in the file is "Query," which is confusing, so as soon as we link the table into our project we rename it to something more sensible, namely Mexico Geometry. (This step is not illustrated).

The result is that now we have a linked table in our project called Mexico Geometry.

If we open this table we see that it has all of the fields from the original drawing plus a field called Geom that contains binary data. Note that the field containing binary data could be called anything. It happens to be called Geom because that's what we named it in the query that created this data from the original drawing. That's a useful name to use for such columns so that right away we know they contain Geometry data if we encounter them in tables.

To create a linked drawing from this table we choose File - Link - Drawing and then in the Link dialog choose This Project () in the Data Source dialog.
In the Link Drawing dialog that follows we choose Table with geometry column in the Type box and the Mexico Geometry table in the Source box. For this example, we have left all of the column boxes in the Columns pane checked. This means that all of them will be imported and available as columns in the drawing's table. However, we don't need either the ID column or the Geom column in the drawing's table. The ID column gives the object IDs for the original drawing.

The column name to use for geometry has already been loaded with the name of the only binary column in the table, Geom, and the type of geometry column has been loaded with Geometry. We press OK.

This creates a new linked drawing, called Mexico Geometry Data in the project. (Importing or linking a drawing from a table or query in the same project will inherit the name of the drawing from the table or query.)
If we open the drawing we see it is just like the original Mexico_eg drawing. It has automatically opened in Latitude / Longitude projection. We can use this drawing to create labels, to create themes, as layers in maps and for just about any other purpose we use a drawing in Manifold.

Let's now do something interesting and create a query that selects only some of the records from our spatial table. We can then create another linked drawing from that query.

We create another query, called Query 2, and write the above text into it to select all records from the Mexico Geometry table for which the SQMI value is greater than 30000. This selects only those records representing larger provinces in Mexico.

To create a linked drawing from the query, we choose File - Link - Drawing and then choose This Project () in the Data Source dialog.
In the Link Drawing dialog we choose **Table with geometry column** as the **Type** and **Query 2** as the **Source** to use. Since we don't need the **ID** field we uncheck it. Press **OK**.

The result is that a second linked drawing, called **Query 2 Data**, appears in our project.
If we open this drawing we see that it contains only the larger provinces in Mexico, those with square mile area greater than 30,000 square miles.

Let us now illustrate the dynamic power of linked drawings by editing the table from which our linked drawing is generated, which will result in changes to the drawing.

We open the Mexico Geometry table and select the record for Durango, a large province in the middle of Mexico.

We press Edit - Delete to delete the record. Even though this is a linked table brought into the project from an external .mdb file, we can delete records in the table because when we linked the table we did not link it in read-only mode. Deleting the Durango record in the linked table actually deletes it in the external .mdb file as well.
If we click on the `Query 2 Data` linked drawing and press **View - Refresh Data** to refresh the drawing, we compel the drawing to refresh itself from the query from which it is created. The query in turn refreshes itself from the linked table and the external `.mdb` file in which the (now altered) data ultimately resides.

We can see that the drawing no longer includes a **Durango** area. The red arrow indicates the empty space where Durango used to be. [Note for the very literal-minded: Manifold does not actually draw red arrows when objects are deleted - the red arrow was added to the illustration.]

If we like, we can open the **Mexico Geometry** table again and click **Edit - Undo** to restore the **Durango** record. This restores it in the external Access `.mdb` file as well.

If we click back onto the `Query 2 Data` linked drawing and press **View - Refresh Data** we see that **Durango** has been restored. Refreshing a linked drawing created from a table that resides in an external data source will refresh the table. Refreshing a drawing linked from a query will recompute the query.

One last thing: if we like, we can use **Edit - Projection** to re-project the `Query 2 Data` drawing into **Orthographic** projection, using the **Suggest** button to choose sensible parameter.
Even though the geometric data in the originating table is in Latitude / Longitude projection, Manifold will re-project on the fly to create the drawing in Orthographic as we have specified.

**Example Scenarios**

Linked drawings created from tables or queries have great flexibility. In addition to the above example, consider these scenarios:

**Scenario 1** - We have a drawing of states and want to create a separate drawing of states for each region. We want to keep all geometric data in the same place so all geometric transforms run in the same context, for example, with the same location precision.

We create a query for each region such as...

```sql
SELECT Geom([ID]), [Region] FROM [States] WHERE [Region] = "East";
```

... and we link a linked drawing from each query. We can then modify the source drawing as desired and thereafter launch a script to update regional drawings so they use the latest data.

**Scenario 2** - We have several different drawings of roads that cover different areas in our area of interest. We want to combine these drawings into a "master" drawing so we can operate upon all roads as a whole. We could create a master drawing using copy and paste, but the drawings are regularly updated and having to repeat the copy and paste process to re-create the master drawing each time a drawing is updated is not very convenient.

We create a query that joins the drawings together in the following form...

```sql
SELECT Geom([ID]) FROM [West]) UNION (SELECT Geom([ID]) FROM [East]);
```

... and we link a drawing from this query. When one of the source drawings is updated, we can update the linked drawing using View - Refresh Data.

**Scenario 3** - We have a drawing of WiFi access points with an operational radius for each access point stored in the drawing’s table. We want to display all WiFi access points in the given area as circles of appropriate size.

We create a query such as...

```sql
SELECT Buffer(Geom([ID]), [Radius], "mi") FROM [Access Points]
```
WHERE EXISTS (SELECT * FROM [Areas] WHERE Contains([Areas].[ID], [Access Points].[ID]));

... and we link a drawing from the query. Note the use of the buffer operator to create circular objects. In this case instead of formatting a point to be larger or smaller we actually create a circular object that is larger or smaller for display.

Scenario 4 - We have an ESRI geodatabase file in .mdb format. We can create a drawing from this geodatabase by choosing File - Import - Drawing and choosing .mdb in the Files of type box to open the .mdb file.

In the Import MDB File dialog we choose Geometry (SHP) in the Type box and press OK. The result is an imported drawing. We could have also used File - Link - Drawing to link a drawing from the geodatabase if preferred.

Note: This too is an artificial example in that if we want to work with ESRI personal geodatabases we can simply import or link to the geodatabase using Database Console. Manifold will automatically detect personal geodatabases and can import or link to them.

Display of Geometry Data within Tables

Tables will display the contents of specialized data types as a descriptive word within angle brackets <> Tables display binary data as <binary>, coordinate system data as <coordinate system> and geometry data as <geometry>

Right clicking on a geometry column head and choosing Format allows us to change the format of how the column is reported. For example, a coordinate system column can be switched between the default format and XML format, which will output XML formatted coordinate system information into each cell. This format is mainly used for debugging or when one wishes to capture a text / XML reporting of coordinate system structure.

For example, if we want to capture the coordinate system from a drawing of Mexico we could write a query called Q as follows:
SELECT CCoordSys(Geom(ID)) INTO X FROM [Mexico Drawing];

Running the above query creates a table called X that has one column containing the coordinate system data.

If we opened table X we could see that by default it reports the coordinate geometry data as `<coordinate system>`.

We can right click on the column heading, choose Format and change the display style from `<default>` to XML.

The result is that each cell reports the coordinate system used using a (lengthy) XML phrase. If we wanted to capture the XML we could click into a cell to edit it, copy the XML and then paste it wherever we wanted.

**Importing or Linking from Spatial DBMS / Exporting to Spatial DBMS**

This topic has discussed the internal mechanics of geometry stored in tables. When drawings are stored in a spatial DBMS there is no need to work directly with geometry columns. Simply use the Database Console to connect to the desired spatial DBMS and then link or import drawings as desired. Manifold will automatically transform the geometry type within the spatial DBMS into Manifold types, either automatically translating the spatial DBMS package’s native geometry type into Manifold Geom or if a Manifold geometry type is already stored into the spatial DBMS using that type to create a drawing directly.

Likewise, when exporting a drawing to a spatial DBMS, if Manifold Enterprise Edition or greater edition is used Manifold will automatically export the necessary geometry data into the DBMS as required.

**Open Data Source Command**

Right clicking a component that is linked from an external data source in the Project pane and then choosing Open Data Source will open the Database Console and connect to the component data source.

Opening a component linked from an external data source in its own window, and then choosing the Open Data Source command in the component menu (for example, choosing Drawing - Open Data Source when a linked drawing window is open) does the same thing.
Invalid or Missing Geometry

At times we might encounter tables or the results of queries that contain invalid geometry. This might happen for a variety of reasons, such as a bug in a third party program that attempts to write geometry or even something as simple as damage to a data set. This is unfortunate, but it would be a shame to lose regular attribute information in such cases, or, even if there is not attribute information it would be a shame not to be alerted to the problem that some objects do not have valid geometry. Manifold will try to salvage what information it can.

Manifold deals with invalid geometry in tables or queries by replacing the invalid metric (that is, the invalid geometry) for each such object with a synthetic metric consisting of a single coordinate located at the \((0,0)\) origin of the drawing's coordinate system. The effect of this is to create such objects with invalid geometry as points at the \((0,0)\) location of the drawing.

For example, if we link a drawing from a table that supposedly contains only areas and we find that some points have appeared at the \((0,0)\) origin of our drawing's coordinate system, we know that some of those areas had invalid geometry. If our drawing is in latitude / longitude, those points will appear all stacked over one another at the intersection of the Equator and the Prime Meridian in the ocean off Africa.

Why does Manifold use the \((0,0)\) location? The points representing objects with invalid geometry have to be put somewhere and this location has the advantage of being easy to remember and, in many common projections, not usually occupied by something else. Experienced GIS users also know that various projection problems will often toss objects into a tiny speck off the coast of Africa so this helps us remember just where Manifold places objects with invalid geometry.

Is it inconvenient to put all points from objects with invalid geometry at the same spot? Not really, since it is easy to select them with a Select Box or other mouse selection method and then we can open the drawing's table, filter on the selected items and see what objects we have at that location.

"Multi-point" geometry values generated by AllCoords and several other query functions are handled the same way: the result of the query function will be a point at the \((0,0)\) origin of the drawing.

When Manifold replaces invalid geometry with a synthetic point metric the IDs of the first ten objects with invalid geometry and a replaced metric will be logged into the History pane.

Limitations with IBM DB2

IBM DB2 Express-C Edition may be used for Enterprise Edition Enterprise Servers without any technical limitations. However, two small usage limitations apply when DB2 Express-C is used for storing geometry in tables within the database to support subsequent concurrent, multiuser editing. When a drawing is linked from a DB2 Express-C data source the following limitations apply:

- Adding a new object and immediately editing it without refreshing the drawing creates an editing conflict.
- Adding a new object and immediately deleting it without refreshing the drawing will fail.

Therefore, when ever editing drawings linked from a DB2 Express-C data source make sure to refresh the drawing after adding a new object before attempting to edit that object or to delete that object.

Notes

Refreshing a linked drawing created from a table that resides in an external data source will refresh the table. Refreshing a drawing linked from a query will recompute the query.

Refreshing a query will refresh all linked drawings descended from that query.

In the example above we could have created a table containing the geometry data in WKB form from the Mexico drawing using the `SELECT … INTO` construction in an action query:

```
SELECT CGeomWKB(Geom(ID)) AS [Geom], [Mexico_eg Drawing].*
```
INTO [WKB Table] FROM [Mexico_eg Drawing];

**Tech Tip**

Remember, ADO.NET connections are read-only. If we are link a drawing from a DBMS table using ADO.NET, we will not be able to edit the drawing. To link a drawing so it is editable we must use some read/write connection technology such as OLE DB or ODBC.

**See Also**

Spatial DBMS
Spatial DBMS Facilities

The above two topics cover the material in this topic from slightly different perspectives and should be read by anyone working with geometry in tables or drawings stored in a DBMS.

Data Storage Strategies
Database Installations
Linked Drawings
Oracle Spatial Facilities
Project Pane - Open Data Source
Queries and Geoms
Spatial Extensions
Virtual Tables for Images and Surfaces

Every image and surface in a Manifold project has a virtual table associated with it. Unlike the tables associated with drawings, a virtual table is not listed in the project pane. It is a "virtual" table in that it does not explicitly appear in the project but nonetheless the fields in it may be used. An image's or surface's virtual table may be used within queries by using the name of the image or surface.

Each pixel in an image or surface corresponds to a record in that image's or surface's virtual table. The virtual table includes the following columns, which are directly analogous to the built-in intrinsic fields of a drawing's table. These fields are the same that are used when an image or surface is copied and then pasted as a table.

- **Center Latitude (I)**: Latitude of the center of this pixel in degrees latitude.
- **Center Longitude (I)**: Longitude of the center of this pixel in degrees longitude.
- **Center Easting (I)**: X coordinate of the center of this pixel in the coordinate system of the image or surface adjusted with the values of the local offset and local scale parameters.
- **Center Northing (I)**: Y coordinate of the center of this pixel in the coordinate system of the image or surface adjusted with the values of the local offset and local scale parameters.
- **Center X (I)**: X coordinate of the center of this pixel in the coordinate system of the image or surface.
- **Center Y (I)**: Y coordinate of the center of this pixel in the coordinate system of the image or surface.
- **Color (I)**: RGB color value expressed as a decimal number (the decimal equivalent to a hexadecimal color value in the form #rrggbb). Read-only for surfaces. Writeable for images.
- **Red (I)**: Red channel value from 0 to 255. Read-only for surfaces. Writeable for images.
- **Green (I)**: Green channel value from 0 to 255. Read-only for surfaces. Writeable for images.
- **Blue (I)**: Blue channel value from 0 to 255. Read-only for surfaces. Writeable for images.
- **Alpha (I)**: Alpha channel value from 0 to 255. Read-only for surfaces. Writeable for images.
- **Invisible (I)**: Boolean: pixel is invisible or visible. Writeable for surfaces. Writeable for images.
- **Selection (I)**: Boolean: currently selected or not. Writeable for surfaces. Writeable for images.
- **Selection Mask (I)**: A byte giving the saved selections mask. Writeable for surfaces. Writeable for images.
- **Height (I)**: For surfaces only: the height of a pixel using whatever data type is used by the surface. Writeable for surfaces.
- **Easting (I)**: X coordinate of the lower left corner of this pixel in the coordinate system of the image or surface adjusted with the values of the local offset and local scale parameters.
- **Latitude (I)**: Latitude of the lower left corner of this pixel in degrees latitude.
- **Longitude (I)**: Longitude of the lower left corner of this pixel in
degrees longitude.

Northing (I)  Y coordinate of the lower left corner of this pixel in the coordinate system of the image or surface adjusted with the values of the local offset and local scale parameters.

X (I)  X coordinate of the lower left corner of this pixel in the coordinate system of the image or surface.

Y (I)  Y coordinate of the lower left corner of this pixel in the coordinate system of the image or surface.

X Offset (I)  X position in pixel coordinates from the upper left corner.

Y Offset (I)  Y position in pixel coordinates from the upper left corner, with the Y axis pointing down.

Writeable fields are only writeable if the queried component is writeable. If the queried component is read-only (for example, if it resides in a read-only .map file or is shared but not checked out from an Enterprise Edition server), then all columns in its virtual table also will be read-only.

The following relationships may help to understand the relationship between different intrinsic columns:

- X Offset (I) equals X (I)
- Y Offset (I) equals Y (I) counted from the top down
- Center X (I) equals X (I) + 0.5
- Center Y (I) equals Y (I) + 0.5

Note: Because the maximum number of records in a Manifold table is 4 gigabytes, images and surfaces that have more than 4 gigabytes of pixels (slightly more than 65500 x 65500) are currently not supported with virtual tables. Their virtual tables do not exist and they cannot be used in queries.

There are three main uses of virtual tables associated with images or surfaces:

- **Storage of Images in Tables** - We can use a query to select all columns for all pixels in an image’s virtual table and then export that query as a table into an .mdb file or to some other database storage. This is a way of storing images within tables in virtually any database.

- **Linked Images, Surfaces or Drawings** - Virtual tables allow us to use the power of spatial SQL to utilize all or part of an image or surface in a query. Other components can then be linked from that query. This allows us to create linked images or surfaces from the query. If the query creates point, line or area objects we can even create linked drawings from the query by linking a drawing to the query.

- **Editing** - Because some columns in virtual tables are writeable (using an update query or a script), virtual tables can be used to edit an image or a surface. The Red (I), Blue (I), Green (I), Alpha (I), Color (I), Selection Mask (I), Selection (I), Invisible (I), and Height (I) columns in virtual tables are writable. Writing to Red (I), Blue (I), Green (I) or Color (I) columns in a virtual table for a palette image can only use colors in the palette for that image. Attempting to write some other color will result in writing a color from the palette that is the nearest match to the attempted color.

- **Analysis** - Virtual tables allow us to analyze images and surfaces with sophisticated SQL statements. For example, virtual tables make it easy to determine the number of pixels occurring within a given height or intensity range, or to create a table of all heights in a surface along with their relative frequencies.

For detailed examples of queries that use virtual tables, see the Queries and Images or Surfaces topic.

**Other Manifold Technologies for Image Storage in Tables**
The topic above discusses image storage in tables either as virtual tables or as the storage of columns in tables that are taken from virtual tables. In addition, Manifold provides two other means of storing images within external databases:

- **Enterprise Edition** - Using Manifold Enterprise Edition we can share images into a Manifold enterprise server. In that case, the DBMS hosting the Enterprise server is simply being used as a big file cabinet and to facilitate collaborative use of such shared image components.

- **Oracle Spatial** - Manifold Enterprise Edition can save images into Oracle Spatial databases using Oracle's GeoRaster technology. This is a very fast technology that scales well for images that are many gigabytes or even terabytes in size. It also allows us to share such images with any other software that is Oracle GeoRaster compatible. See the Oracle Spatial Facilities topic.

**See Also**

Linked Drawings
Geometry in Tables
Spatial Extensions
Linked Images
Linked Surfaces
Queries and Images or Surfaces
Raster Extensions
Active Columns

Active Columns are columns in tables that show the results of a script function. Active Columns are used to show automatically computed values in each cell of the column, similar to how a spreadsheet is used to automatically show computed values for a formula in a cell. This makes it easy to create columns in tables where the values are, for example, computed from other columns in the table.

Active columns might be used in a simple way, for example, to show values given in English height measurements within a table as Metric height measurements (using a simple multiplication by a conversion factor), or they could involve considerably more sophisticated computations using table look-ups or even specialized programming objects. Use Table - Add - Active Column to add an active column to a table. Creating an active column creates a new, special script component that is bound to the table.

Each active column is assigned a script function within the new script component. The script component can contain other functions and subprocedures that are not displayed in any active column. For example, common subprocedures used by more than one active column could be written into the script component. Global variables and other elements of scripts can also be written into the script component for use by any active column in that table. The type and size (for types like fixed-length ANSI strings that require a size) of an active column are set by the script/active column author.

Use of active columns requires the ability to write functions in an ActiveX scripting or .NET language. For example, simple functions are very easy to write in Visual Basic Scripting and may be created by non-programmers. For more advanced use, any ActiveX scripting language or any .NET language supported on the system may be used. For example, the script that powers an active column may be written in Jscript, the Microsoft implementation of Javascript.

To create an Active Column:

1. Open the table.
2. Right click on a column head and choose Add - Active Column or choose Table - Add - Active Column from the main menu.
3. Specify the function name, the name to appear on the column head and the data type returned by the function.
4. Make sure the Compute option is not set to on the fly. Press OK.
5. A script component will be created under the table and opened for editing. Write the function to be used by the Active Column. Close the script component.
6. Once the script and active column are verified to work correctly, if desired the computation mode may be changed to on the fly. Use of on the fly mode will prevent most script editing operations in the script window.

Active Columns can contain programs written in any ActiveX scripting language or any .NET language available in your system.

Add Active Column Dialog

<table>
<thead>
<tr>
<th>Function</th>
<th>Name of the function called by this Active Column.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Language</td>
<td>Choose an ActiveX or .NET language in which the Active Column script will be written.</td>
</tr>
<tr>
<td>Add references for standard .NET modules</td>
<td>Adds references to standard .NET modules for active columns written in .NET languages.</td>
</tr>
<tr>
<td>Name</td>
<td>Name of the Active Column in the table.</td>
</tr>
<tr>
<td>Type</td>
<td>Data type returned by the function and hence the type of the column in the table. If the called function returns a type other than this, Manifold will try to convert the data into this type.</td>
</tr>
<tr>
<td>Size</td>
<td>Specifies size for data types (such as fixed length text types) requiring a size.</td>
</tr>
</tbody>
</table>
Compute  Computation method to be used for this Active Column:

- **on demand** - Compute the column when the table is first opened and cache the values. Thereafter, recompute values on a user Recompute command.

- **on demand, recompute after changes to script** - Same as on demand, but also recompute values if the script is edited.

- **on the fly (no cache)** - Always recompute cells dynamically whenever necessary without ever caching values. The slowest mode, but always up-to-date. Use of this mode will not allow most editing of scripts. To edit a script associated with an active column, first change the Compute mode to some other mode, edit the script as desired, verify it runs and then change the Compute mode back to on the fly.

- **on user request** - Recompute cell values only when the user issues a Recompute command. The fastest mode and the default.

Once formula results are computed for the table, the Active Column may be used like any other column. It may be used for thematic formatting or in an SQL expression, for example. The table may be sorted by clicking on the Active Column head.

**Computation Controls**

Manifold provides a Compute control to specify how the Active Column should be computed. The Compute parameter is specified when the Active Column is created. It may be changed at any time by right clicking on the Active Column head and choosing Edit to launch the Active Column dialog again. Different Compute modes are provided to allow user control over when the script in the Active Column is executed to update the values in the column.

Although Active Columns are conceptually similar to spreadsheet cells in that the values are intended to be automatically computed, the formulas used within Active Columns can often be very large and complex scripts (large programs consisting of many subroutines). In the case of drawings containing many objects, and thus many records in the drawing's table, it could take a long time to recompute the Active Column for all records. Accordingly, unlike spreadsheets that are set by default to automatically recompute every cell when anything in the spreadsheet is changed, the default behavior of an Active Column is to recompute the values in the column only when the Recompute command is issued by the user, the on user request setting.

When an Active Column is recomputed using the on user request or the on demand settings the values in the Active Column will be cached, that is, held unmodified in the column, even if values upon which they depend from other columns have been changed. This can be changed by using the on the fly setting, which results in behavior similar to that of spreadsheets where cells are always automatically recomputed.

If caching is turned off for on the fly recomputation, any such action using the Active Column will cause a dynamic recomputation of the values in the column. For simple formulas the difference between on the fly and other modes is very slight; however, for more complex functions or major programs written into an Active Column the difference is substantial. In such cases the on user request mode should be used.

The Recompute command is highly optimized to avoid unnecessary context switches when refreshing values in cacheable Active Columns. This makes the Recompute command the fastest way to compute values in Active Columns. Invoking the Recompute command on an Active Column whose function can not be found will display an error message when the script cannot be executed.

**Tech Tip**

Why does on the fly mode not allow editing of the active column's script? On the fly mode causes the active column, and thus the script, to respond to any changes. Suppose we attempt to edit a script's text or even to enter the initial text: when we make an edit, the change in text (even just one character) will be detected by the system as a change. In response the table window will attempt to compute the values for any active column cells that are visible; however, it is quite likely that the script as changed is not yet functional. The script engine
therefore signals an error and the script window selects the text it thinks is incorrect (which might be just a single character the user has just typed).

Avoid this restriction on editing by using one of the following techniques:

- Create an active column with another computation mode, make sure it works and then switch computation mode to "on the fly."
- Avoid execution of active column code by making the active column invisible, that is, by hiding it in the table window or by closing the table window itself for the editing period.

**Example**

We will open the **Order Details** table from the **Nwind.mdb** sample database and add an active column to it.

We've opened the table and have hidden all fields except **Order ID**, **Unit Price** and **Discount**.

To create an Active Column we choose **Table - Add - Active Column** from the main menu.

In the **Add Active Column** dialog we enter **Discounted** for the name of the function to use, **Price with Discount** as the name of the column and we choose **Floating-point (double)** for the type of our column. For this example we will choose **on user request** for the computation method. Press **OK**.
A script component called **Order Details Script** is created in the project pane underneath the **Order Details** table. If this is the first Active Column, the script component is popped open for editing in a script window using default template code.

```
Function Discounted
    Discounted = Nothing ' Record.Data("Column")
End Function
```

The template code is not a functioning Active Column script - it simply reminds us what the structure of a Active Column script should be.

```
Function Discounted
    Discounted = Record.Data("Unit Price") * (1 - Record.Data("Discount"))
End Function
```

We edit the script component to add the above text. Note how script windows in Manifold will color the different parts of a script to help us keep track in our coding. Advice to newbies: pay careful attention to details and check for keyboarding errors. "Unit Price" is not the same thing as "UnitPrice".

<table>
<thead>
<tr>
<th>Order...</th>
<th>Unit Price</th>
<th>Discount</th>
<th>Price w...</th>
</tr>
</thead>
<tbody>
<tr>
<td>10000</td>
<td>27</td>
<td>0</td>
<td>27</td>
</tr>
<tr>
<td>10000</td>
<td>22</td>
<td>0</td>
<td>22</td>
</tr>
<tr>
<td>10000</td>
<td>21.35</td>
<td>0</td>
<td>21.35</td>
</tr>
<tr>
<td>10003</td>
<td>43.7</td>
<td>0.05</td>
<td>41.9145...</td>
</tr>
<tr>
<td>10004</td>
<td>85</td>
<td>0</td>
<td>85</td>
</tr>
<tr>
<td>10004</td>
<td>30.7</td>
<td>0</td>
<td>30.7</td>
</tr>
<tr>
<td>10005</td>
<td>34</td>
<td>0.15</td>
<td>28.9000...</td>
</tr>
<tr>
<td>10006</td>
<td>4.2</td>
<td>0</td>
<td>4.2</td>
</tr>
<tr>
<td>10006</td>
<td>11.3</td>
<td>0</td>
<td>11.3</td>
</tr>
<tr>
<td>10008</td>
<td>7</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>10008</td>
<td>37.1</td>
<td>0</td>
<td>37.1</td>
</tr>
</tbody>
</table>

When we close the script window an Active Column appears. Right click on the **Price with Discount** column head and choose **Recompute**. The Active Column uses a different background color to indicate it contains computed values. Foreground and background colors for Active Columns may be set in the Tools - Options dialog. The values it shows are those computed using the function we wrote.
We can widen the column to see the full name we specified. We can right click on the column and choose Format to format the columns to only two digits after the decimal point. To change the column so that it re-computes automatically, we right click on the Price with Discount column head, choose Edit and change the Compute value to on the fly.

Syntax and Conventions for Active Columns

When used within an Active Column script, ‘Record’ refers to the context record that has the following properties:

- ID: The ID of the drawing object. Only for drawing tables.
- SequentialNumber: Sequential number of the record within the table.
- Data: A collection of values within a given record.
- DataText: A collection of values with a given record represented as formatted strings.

Active Column scripts can reference other components and can even reach any other scriptable objects installed within the system such as Microsoft Office applications.

Context Menu Commands for Active Columns

Right click on an Active Column head to call up a context menu of commands. In addition to the usual table column commands, Active Columns have:

- Edit: Edit the definition of the Active Column using the Edit Active Column dialog (functionally equivalent to the Add Active Column dialog).
- Edit Script: Enabled for Active Columns. Edit the script associated with this column.
- Flatten: Enabled for Active Columns, Rank Columns and Relations. Convert the column to a regular column by using the current values in the cells.
- Recompute: Recompute the values in an Active Column.

Debugging

If you have purchased the Debugger option for Professional Edition or if you have licensed Enterprise Edition (which includes the Debugger) you can use the Debugger to debug Active Column scripts written in ActiveX languages that support Microsoft specifications for debugger interfaces. Scripts associated with Active Columns can run in the debugger. To debug an Active Column script, place one or more breakpoints within the script, right click on an Active Column cell and choose Run under Debugger from the pop up menu. Doing this runs the function while generating values for the Active Column in the context of the clicked cell. The resulting value will be discarded. All regular runs of Active Column functions are done without the debugger.
Converting to Regular Columns

Right click on an Active Column head and choose Flatten to convert the column to a regular data column. The current value of each cell will be used to populate the regular column.

Notes

Active columns can only reliably refer to non-active columns. Using one active column from another will not force re-computation and will either return the last cached value or no value at all. A "no value" return is the value with a 'no value' flag that is supported by all ActiveX scripting hosts.

Choosing an unexpected type for the column in the Add Active Column dialog will force a conversion to that type if possible and will return no values if a conversion is not possible.

We can manipulate drawings from Active columns. The following Active column operates on a drawing's table and returns the number of points in the first branch of the object.

```ironpython
Function PointsInFirstBranch
    Set objSet = Table.Owner.ObjectSet
    Set obj = objSet(objSet.ItemByID(Record.Data("ID")))
    PointsInFirstBranch = obj.Geom.BranchSet(0).PointSet.Count
End Function
```

IronPython scripts intended to be used for active columns should only include the body for a single script function.

See Also

See the Scripts topic for information on scripting. See Active Columns using VBScript and Active Columns using Jscript for examples using active columns and script functions written in VBScript and Jscript.
Finding Data in Tables

There are several ways of finding data in tables. Some methods of finding data involve making selections.

- Find / Find Next commands may be used to search for a value or a regular expression that is a pattern. Searches can be restricted to only one field if desired.
- Manual sorting using the Sort Ascending or Sort Descending context menu commands from the table context menus can be used to sort the table by a column. The table can then be manually scrolled to find the records desired.
- Tables may be sorted by clicking on a column head to sort by that column or by using the View - Sort dialog to sort records using several fields at once.
- An Active Column can be created to compute values by sophisticated rules. The table can then be sorted by the Active Column to bring desired ranges of values generated by the rule to the top of the table.
- More Like This commands are a fast way of finding more data like a particular record or the contents of a particular cell.
- The Decision Support System provides a sophisticated means of finding records through flexible criteria.
- Queries may be used to create classic database queries that find and display records from tables using Structured Query Language (SQL).
- The Query Toolbar provides a fast means of using simple queries to select records in tables.
- ViewBots can be used to find data in tables by selecting the records picked out by the ViewBot.
- If the table is a drawing's table, selection operations in the drawing can be used to find records in the table that are otherwise impossible to distinguish. For example, using Select Circle to select all points in a drawing that fall within a particular cluster will also select the records associated with those points in the drawing's table. See the Selection topic.

If we are looking for a specific value the fastest way is usually to use Find. If we are looking at a record that is similar to those we want we can use the More Like This system. For browsing, tables may be sorted either by columns or by using Sort and browsed by scrolling through the table. This can be a remarkably fast approach even for large tables.

When working with large tables it is often convenient to open the table in two windows at once. One of the windows is used to make selections and the other window is used to show just the selected records.

Press in the Filter Selected filter button in one of the table windows to show only selected records in that window.

For example, in one window we might use More Like This Cell (Sorted) to sort records by their similarity to the value in a particular cell. We can then add the top five or ten records to the selection and see them appear in the table window that is filtered to show only the selection. We can then click on the full table window and do a different procedure to add to the selection. In this way we build up a list of selected records in the

Once we find the data we want in a table we might want to make a selection of interesting records and then save that selection in the Selections pane. We might also want to use bookmarks to note records.
More Like This

Manifold's More Like This commands operate within tables to help us find records that are similar to those we see. They provide a shortcut way of using Manifold's Decision Support System in a rapid, one-click way.

More Like This Cell / More Like This Cell (Sorted)

To find more records that have similar values to a given cell, right click on the cell and choose More Like This Cell from the context menu. Other records that have a value "like" that will be selected. Choosing the (Sorted) version of the command will sort the table so that records with values in this cell most like the target cell will be placed at the top of the table and those records that are least like the target cell will be placed at the bottom of the table.

The More Like This system ranks exact matches the highest, and then ranks records in decreasing similarity based on a flexible comparison to the target cell. For example, suppose we open the Customers table in the Nwind sample database and try to find more records that have "USA" in their Country field. If we right click onto a "USA" cell and choose More Like This Cell (Sorted) the table will be sorted so that "USA" records come first, followed by a "usa" record and then a series of "UK" records. After that, the table sort will be so "unlike" the specified "USA" cell that the sort order will not be very helpful. However, all the "USA" and "UK" records will be at the top.

More Like This Record / More Like This Record (Sorted)

Right clicking onto a record handle and choosing More Like This Record will find additional records that are "like" that record. The (Sorted) version of the command will sort the table so that records most like the given record are at the top and those records least like the given record will be at the bottom.

The command works by examining all the fields that are displayed in the table and launching a Decision Support System flexible query that seeks to find records that have values in their fields like those that are displayed in the given record. Columns that are hidden will not be considered.

More Like This Record (Sorted) Example

Suppose we open the Customers table in the Nwind.mdb database in two different table windows.

In the window shown on the left in the illustration above we've hidden all fields except CustomerID and ContactTitle. In the window on the right we've hidden all the fields except CustomerID and Country. In both cases, we will right click on the "BLAUS" customer ID record and choose More Like This (Sorted).

For the table on the left we are therefore looking for more records that have "BLAUS" for customer ID and "Sales Representative" for the contact title. For the table on the right we are looking for more records that have "BLAUS" for customer ID and "Germany" for the country.
The result on the left found records that were like the one indicated considering that only the **CustomerID** and **ContactTitle** columns were displayed in the table. Unsurprisingly, the top records in the sorted table all have "Sales Representative" as the value for the **ContactTitle** field. Because "BLAUS" is a unique value for **CustomerID** the **More Like This** consideration of this field places that particular record at the top of the table.

The table on the right shows the result of a **More Like This** search on the same record. However, in this case only the **Country** field was displayed so the **More Like This** reckoning looked for records that also had "Germany" for country.

**More Like... Dialog**

The **More Like** dialog allows custom control over which fields are reckoned in the **More Like This** query to find records that are "like" the target record.

**To Use the More Like... Dialog:**

1. Right click on the record handle for the target record. Choose **More Like...** from the context menu.
2. All columns displayed in the table will be listed in the dialog. For each column that is to be used, double click into the **Similarity** cell and choose how that column should be used (from "slightly similar" to "extremely similar" to find possibly similar records.
3. In the **Select** box, choose which records are to be selected. The choice is either all those records that rank higher than a given percentage or those records that are in the top given number of records.
4. Check the **Sort results by rank** box if the table should be sorted by the **More Like** ranking.
5. Press **OK**.

The **More Like This** system functions by creating a Decision Support System query within Manifold based on the columns and similarity criteria desired and then ranking all the records in the table by how well they match that DSS query. Records are ranked as a percentage from 0 % (not at all like the target record criteria) to 100 % (identical to the target record criteria).

The **Similarity** choices allow specification of how similar the selected fields will be to the target record's fields. The default choice is normally **similar**. Choose other **Similarity** options from the list box to force stronger or weaker similarity for that field.

**More Like... Dialog Commands**

- **Column**: Column names that are currently enabled for display in the table.
- **Similarity**: Double click into the **Similarity** cell for each column that is to be used in the search.
- **Select ranks higher than ... %**: Select all records whose DSS
similarity ranking is greater than the given percentage.

top … records - Select this number of the highest ranking records.

Sort results by rank - If checked, sort the table by the DSS ranking of each record for the given criteria.

**Similarity Options**

- **slightly similar**: Weakest similarity required. Equivalent to using an Is Slightly hedge in the DSS query. Would require a 12.5% ranking to get the same effect as a 50% ranking achieved with similar.

- **somewhat similar**: Weaken similarity required. Equivalent to using an Is Somewhat hedge in the DSS query. Would require a 25% ranking to get the same effect as a 50% ranking achieved with similar.

- **similar**: Use a standard DSS profile to rank desirability of values.

- **very similar**: Strengthen similarity required. Equivalent to using an Is Very hedge in the DSS query. Would require a 70% ranking to get the same effect as a 50% ranking achieved with similar.

- **extremely similar**: Strongest similarity. Equivalent to using an Is Extremely hedge in the DSS query. Would require a 79.4% ranking to get the same effect as a 50% ranking achieved with similar.

**Notes**

The screen shots for this topic were done in a "live" Manifold session with two table windows open at the same time into the Customers table. Note that different windows into the same table can have different arrangements of columns visible and can be sorted independently.

See the Tables - Context Menus topic for a reference listing of context menu commands in tables.

The More Like… dialog creates and launches an internal Decision Support System query. This is a simple way of using DSS without learning about profiles or queries. For those interested in the details, the similar choice for Similarity is equivalent to using a criterion profile in the DSS query that (for numeric fields) uses a Gaussian bell curve about one standard deviation wide to either side of the target value. Choosing other Similarity options is the same as using hedges in the DSS query to make the Gaussian bell curve wider (weaker similarity) or narrower (stronger similarity).

The internal profile is created by considering the actual values that occur throughout the table. This assures the target More Like This profiles are a reasonable match to the actual data in the database.
Rank Columns

Rank Columns are used to construct a Decision Support System query and to report the results. A rank column reports percentages. The Manifold Decision Support System (DSS) provides a way to select records using flexible criteria with queries. DSS results are reported as percentage values from 0 to 100 percent within a rank column in a table. Creating a rank column creates a DSS query, the results of which are reported in the column.

Instead of using fixed criteria such as "greater than 50" in a query, DSS can handle a situation where the criterion might be more than 50 most of the time but that sometimes a value near 50 is also acceptable. The science of using flexible criteria is generally referred to as "fuzzy" logic. See the Introduction to Decision Support topic for more information on fuzzy logic.

Decision Support describes objectives for each data field using criteria. Criteria are curves that show which values for the data field are more desired and which are less desired. One or more criteria may then be used in a ranking query to find records that are a good fit to the combined requirements of all the criteria used. The results will be reported using a rank number, a percentage from 0% to 100% that appears within a Rank Column in the table. Records that fit the query well will be ranked at or near 100%. Records that fit the query less well will be ranked at increasingly lower percentages.

Records may then be organized by clicking on the Rank Column heading to sort the table in order of DSS ranking.

To select records using Decision Support:

1. Open the table of interest.
2. Use Table - Add - Rank Column to add a Rank Column to the table.
3. In the Add Rank Column dialog, specify the name of the column.
4. Press New to add a new criterion to the Criteria pane.
5. In the Add Criterion dialog, choose the name for the criterion, the field (column) to be used and the type of criterion. Types such as High and Low provide presets that are automatically computed based on the data in the table. Press Apply to see the effect of the Type and press OK to create the criterion.
6. Continue adding criteria using steps 4 and 5 above for all the fields you wish to use. Criteria appear organized in a hierarchical diagram under the names of the fields they use.
7. Create a query by dragging and dropping criteria from the upper pane into the lower pane. Change the Junction from Or to And as desired by double clicking onto the Or. Place a Not in front of a criterion if desired by double clicking into the Not column and choosing Not.
8. Press OK and a Rank Column giving DSS values for all the records will appear in the table. Sort the column by clicking on the column head to see the records displayed in rank order. Records with a rank value of 100% most closely match the rank query.

Another way to make a selection using DSS is to use a ViewBot in the ViewBots pane. For example, suppose we have created a Rank Column named Ranking. We could add a ViewBot that reported for the Ranking field the records Greater or Equal To .90. This would report all records where the total Ranking was greater than 90%. We could then click Replace Selection in the ViewBots pane to select those records. Note that we use a value of .90 in the ViewBot since rankings are percentages.

Percentages are numeric values from 0 to 1.00 that are normally multiplied by 100 and followed by a percent sign for reading convenience. For example, an 85% value shown in a rank column is actually the number .85.

Once created for any rank column all the criteria in the Add Rank Column pane will be available for use in any other rank column in that table.

Rank columns can only be created using numeric columns that are native to the table, intrinsic or foreign columns. ["Foreign" columns are columns mapped from other tables through relations.] They can not be created using other rank columns or using Active Columns. Rank columns can be created with tables imported into the project or linked into the project from external files or providers.

Converting to Regular Columns

Right click on a Rank Column head and choose Flatten to convert the column to a regular data column. The current value of each cell will be used to populate the regular column.
See Also

Decision Support System - for more detailed topics on using DSS in Rank Columns
Rank Columns / Decision Support System - a step-by-step example of creating a rank column and using DSS.
Add Rank Column Dialog - for details on options within this dialog.

Editing Tables

Editing Data in Tables
We may edit data in tables interactively by clicking into individual records, by using Find and Replace and by making global edits using the Transform toolbar. Editing data in tables that have been imported into Manifold will change the data only within the Manifold project. Editing data within external tables that have been linked into a project will change the data in the external table.

We can change fields in records by double-clicking on the field and entering a new value. Press Enter to apply the new value. If we change our minds after clicking a field, we can hit the Esc key to abandon the edit. Some fields in some databases may be limited to certain values, in which case choice boxes or other dialogs will pop up as necessary to provide allowed choices.

For "global" edits that affect an entire column throughout the entire table or selection use the Transform toolbar. Most table editing commands will be accomplished by a series of steps using the transform toolbar.

To edit individual records, we can use standard Windows power moves as set forth below:

Editing Cells

The contents of cells may be edited by double clicking into the cell, by pressing Enter or by clicking on the cell to highlight it (or by moving the highlighting using arrow keys) and pressing F2. When finished editing, press Enter to accept the value or click with the mouse anywhere outside the cell to accept the value.

Cells will only accept values that are compatible with the data type of that field. For example, double clicking into an integer field and attempting to enter the value 425 Main Street will result in no changes made on pressing Enter. Attempting to enter a floating point value such as 3.1415 into an integer cell will result in only the integer part of the value (3) being entered on pressing Enter.

Cells that are formatted in a given column formatting style will accept edits in any compatible style. For example, if a date and time column is formatted to show dates as 3/09/2001 and one enters 9 mar during the year 2001 the entry will be accepted and shown as 3/09/2001. In this case, Manifold knows that "mar" in date and time fields means March. If the or time is not explicitly given the system uses the current system year and time. To specify all parts of the data and time value, enter explicitly those parts desired. If one enters 9 mar 99 the entry will be accepted and shown as 3/09/1999.

Editing the value of a fixed-length text column in a table window limits the length of the editable text to the maximum length accepted by the column.

Right clicking onto a cell brings up a context menu of choices. Note that the Delete command will reset the value of the clicked cell to the default value for that column. For example, using Delete on a cell in an integer column will reset the value of that cell to 0.

Adding Records to a Table

Tables that are associated with drawings cannot have records added to them from the table window since each record in the table is associated with an object in the drawing. Adding a record means adding an object: this is done in either a drawing window or a map window (if the drawing is used as a layer in a map) by inserting a new point, line or area object. When the new object is inserted into the drawing a new record for it will appear in the table.

To insert new records into tables that are not associated with drawings, click into cells in the new record row (marked with an asterisk) to edit their values. Press SHIFT-Enter to append the new record to the table.
Suppose we start with a table with only two records.

If we click on the first empty cell in the new record row and press **Enter** we can begin entering a new name. In this case, we have typed **Ben**.

Pressing **Enter** will enter that value into that cell. Note that the status icon in the row handle has changed to a small pencil to show we are entering a new record. We can press the right arrow on the keyboard to move to the next cell in the row.

If we press **Enter** we can begin editing that cell. After entering the text **Franklin** we can press **Enter** once more to have the cell accept the new value.

If we are happy with our new record we can right click on the record handle (editing pencil icon) or on the new cells and choose **Append** to append the new record to the table. Choosing **Cancel** will cancel the operation and choosing **Clear** will clear all values from cells so we can begin again. Keyboard shortcuts: use **SHIFT-Enter** to append and **SHIFT-Backspace** to cancel.
If we append the record a new record is added to the table. Note that the editing pencil icon has been replaced with a current record icon in the usual way. A new record row appears as well.

**Changing Fields in Many Cells at Once**

When changing the value of a specific cell, that field will be changed to that value for all selected records:

<table>
<thead>
<tr>
<th>Accounting Manager</th>
<th>Rue de</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sales Representative</td>
<td>Carrier</td>
</tr>
<tr>
<td>Sales Representative</td>
<td>City, Ci</td>
</tr>
<tr>
<td>Sales Associate</td>
<td>8 John</td>
</tr>
<tr>
<td>Marketing Manager</td>
<td>Garden</td>
</tr>
<tr>
<td>Sales Associate</td>
<td>Maube</td>
</tr>
<tr>
<td>Sales Representative</td>
<td>67, av</td>
</tr>
<tr>
<td>Sales Manager</td>
<td>1 rue</td>
</tr>
<tr>
<td>Marketing Assistant</td>
<td>1900</td>
</tr>
</tbody>
</table>

Suppose we wish to change the title to "Sales Person" in all selected records.

<table>
<thead>
<tr>
<th>Accounting Manager</th>
<th>Rue de</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sales Person</td>
<td>Carrier</td>
</tr>
<tr>
<td>Sales Person</td>
<td>City, Ci</td>
</tr>
<tr>
<td>Sales Person</td>
<td>8 John</td>
</tr>
<tr>
<td>Marketing Manager</td>
<td>Garden</td>
</tr>
<tr>
<td>Sales Person</td>
<td>Maube</td>
</tr>
<tr>
<td>Sales Person</td>
<td>67, av</td>
</tr>
<tr>
<td>Sales Person</td>
<td>1 rue</td>
</tr>
<tr>
<td>Marketing Assistant</td>
<td>1900</td>
</tr>
</tbody>
</table>

Press **Enter** or double click into the field in any of the selected records and enter "Sales Person" and then press **Enter**.

The value will be changed in that field in all of the selected records.

This above behavior can be turned off by unchecking the **Apply table cell editing to all selected records** box in the Tools - Options dialog.

**Keyboard Editing**

One may also edit data in Table windows using "keyboard style" data editing. **Double-click** into a field to edit it and then push **Enter**. Move the edit box to another field by using the keyboard cursors. **Push Enter** to open the field for editing, enter the new value, and then hit **Enter** to apply the value.

The **Tab** key will move one cell to the right of the active cell. A **Shift-Tab** will jump one cell to the left of the active cell.

Keyboard editing is a rapid way of moving the cursor through columns of data and making changes as desired.
Finding and Replacing Data

Data in tables can be found and, if desired, replaced with the Edit - Find, Edit - Find Next and Edit - Replace dialogs. These are very similar to the analogous dialogs in Microsoft Office applications. See the Finding / Replacing Data topic. For details on options in the Find and Replace dialogs, see the Edit - Find / Find Next and Edit - Replace topics.

Note: Manifold Find and Replace operate from the current record position forward to the end of the table. To being a search at the beginning of the table, press the Restart dialog in the Find or Replace dialogs.

Copying and Pasting Columns

Entire columns of data may be copied and pasted or copied and appended as new columns into tables.

Consider a table containing columns of data like the above. We would like to copy the contents of the SQMI column and paste it into another column. To do so, we first right click onto the column head for the SQMI column.

In the resulting context menu, we choose Copy.

Suppose we would like to paste the contents into the Square Miles column: we right click that column header...
... and in the resulting context menu we choose Paste.

<table>
<thead>
<tr>
<th>ID</th>
<th>SQMI</th>
<th>Square Miles</th>
<th>AUTO</th>
<th>BUSES...</th>
<th>Place_name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>63068</td>
<td>63068</td>
<td>88881</td>
<td>1373</td>
<td>SAN LUIS POTOSI</td>
</tr>
<tr>
<td>2</td>
<td>73252</td>
<td>73252</td>
<td>52567</td>
<td>1400</td>
<td>ZACATECAS</td>
</tr>
<tr>
<td>3</td>
<td>125181</td>
<td>125181</td>
<td>68060</td>
<td>1401</td>
<td>DURANGO</td>
</tr>
<tr>
<td>4</td>
<td>192050</td>
<td>192050</td>
<td>172500</td>
<td>3004</td>
<td>SONORA</td>
</tr>
<tr>
<td>5</td>
<td>58328</td>
<td>58328</td>
<td>121750</td>
<td>3111</td>
<td>SINALOA</td>
</tr>
<tr>
<td>6</td>
<td>79384</td>
<td>79384</td>
<td>281275</td>
<td>3536</td>
<td>TAMAUHULAS</td>
</tr>
<tr>
<td>7</td>
<td>64524</td>
<td>64924</td>
<td>352345</td>
<td>4857</td>
<td>NUERO LEON</td>
</tr>
<tr>
<td>8</td>
<td>69201</td>
<td>69201</td>
<td>441081</td>
<td>2549</td>
<td>BAJA CALIFORNIA</td>
</tr>
<tr>
<td>9</td>
<td>73475</td>
<td>73475</td>
<td>45747</td>
<td>375</td>
<td>BAJA CALIFORNIA</td>
</tr>
<tr>
<td>10</td>
<td>69201</td>
<td>69201</td>
<td>441081</td>
<td>2549</td>
<td>BAJA CALIFORNIA</td>
</tr>
<tr>
<td>11</td>
<td>149982</td>
<td>149982</td>
<td>234115</td>
<td>4277</td>
<td>COAHUILA</td>
</tr>
</tbody>
</table>

The values from the SQMI column will be pasted into the Square Miles column.

Suppose that instead of choosing Paste in the context menu, we had right clicked onto Square Miles and had chosen Paste Append?

<table>
<thead>
<tr>
<th>ID</th>
<th>SQMI</th>
<th>Square Miles</th>
<th>AUTO</th>
<th>BUSES...</th>
<th>Place_name</th>
<th>SQMI 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>63068</td>
<td>0</td>
<td>88881</td>
<td>1373</td>
<td>SAN LUIS POTOSI</td>
<td>63068</td>
</tr>
<tr>
<td>2</td>
<td>73252</td>
<td>0</td>
<td>52567</td>
<td>1400</td>
<td>ZACATECAS</td>
<td>73252</td>
</tr>
<tr>
<td>3</td>
<td>125181</td>
<td>0</td>
<td>68060</td>
<td>1401</td>
<td>DURANGO</td>
<td>125181</td>
</tr>
<tr>
<td>4</td>
<td>192050</td>
<td>0</td>
<td>172500</td>
<td>3004</td>
<td>SONORA</td>
<td>192050</td>
</tr>
<tr>
<td>5</td>
<td>58328</td>
<td>0</td>
<td>121750</td>
<td>3111</td>
<td>SINALOA</td>
<td>58328</td>
</tr>
<tr>
<td>6</td>
<td>79384</td>
<td>0</td>
<td>281275</td>
<td>3536</td>
<td>TAMAUHULAS</td>
<td>79384</td>
</tr>
<tr>
<td>7</td>
<td>64524</td>
<td>0</td>
<td>352345</td>
<td>4857</td>
<td>NUERO LEON</td>
<td>64524</td>
</tr>
<tr>
<td>8</td>
<td>69201</td>
<td>0</td>
<td>441081</td>
<td>2549</td>
<td>BAJA CALIFORNIA</td>
<td>69201</td>
</tr>
<tr>
<td>9</td>
<td>73475</td>
<td>0</td>
<td>45747</td>
<td>375</td>
<td>BAJA CALIFORNIA</td>
<td>73475</td>
</tr>
<tr>
<td>10</td>
<td>69201</td>
<td>0</td>
<td>441081</td>
<td>2549</td>
<td>BAJA CALIFORNIA</td>
<td>69201</td>
</tr>
<tr>
<td>11</td>
<td>149982</td>
<td>0</td>
<td>234115</td>
<td>4277</td>
<td>COAHUILA</td>
<td>149982</td>
</tr>
</tbody>
</table>
In that case, a new column called SQMI 2 would have been created and the values from the SQMI column would have been pasted into that new column.

The above example shows copy and paste within the same table; however, Manifold also supports copy and paste of binary data in table columns through the Windows clipboard when other applications are used. For example, we can copy a column from an Excel spreadsheet and paste it into a Manifold table column.

**Using Regular Expressions**

Regular expressions are a concise and flexible notation for finding and replacing patterns of text and are used in the Find and Replace dialogs.

See the Regular expressions topic for information on using regular expressions. These are a much more sophisticated and flexible method than simply using the * and ? characters.

**Tech Tip: Changing Ordinary Fields in Tables does not Move Objects**

The positions of objects in drawings are embedded within the geometry of the drawing without need to have latitude and longitude values listed in a table. Sometimes we create drawings from geocoded tables that have a latitude and longitude for each record. We might wish to have the latitudes and longitudes that were used to position the points immediately at hand in the drawing’s table and so when pasting the geocoded table as a drawing we may tell Manifold to bring along the latitude and longitude fields along with the other fields.

Whatever the motive or process that was used to create a drawing, it is often the case that "latitude" and "longitude" numbers will be copied to fields in a table as well as being embedded within the geometry of a drawing. GIS beginners often mistakenly think that the positions of items in the drawing are determined from the listings of latitude or longitude fields in the table, so that editing these fields will move the objects. That's not the case.

We can use intrinsic fields to display in a table the actual latitudes and longitudes of objects associated with records. However, intrinsic fields are special, system-generated fields and are different than "latitude" and "longitude" fields we might bring into a table that are ordinary text or numeric fields. The rest of this topic explains why this is so.

Drawings are often created in Manifold from geocoded tables. A geocoded table is a table where each record has latitude and longitude fields that specify the location of that record.

Suppose we have an Access database that contains a list of towns together with their latitude and longitude coordinates:
We can import this Access table into a table in Manifold (either by importing the data or by linking to the external table):

To create a drawing from this table we use Edit - Copy to copy the table and then we use Edit - Paste As - Drawing in the project pane to create a new drawing based on the data in the table. Each geocoded record will appear as a point in the drawing.
We can combine this new drawing in a map together with a base map of the United States to see at a glance where the various points are located.

When we use the Edit - Paste As - Drawing command to create a new drawing, Manifold will automatically create the associated data attribute table for that drawing. By default, that new table will have at least one field, the object ID field. During the Paste As operation Manifold will give us the option of copying additional fields from the original table into the new data attribute table for the drawing. The dialog presents a list of check boxes for each field in the original table. We can copy any of the fields we desire.

For example, if our geocoded table is a list of customer records containing fields such as name, address, city, state, and so forth together with a latitude and longitude for each record, we might wish to copy the name, address and other informational fields. There is no particular technical reason to copy the latitude and longitude fields (these fields in the original table will be used automatically to position the points in the drawing for each record) but we can do so if for some reason we want to have this information in the table.

If we do copy the latitude and longitude fields into a table associated with the drawing, we will see them in the table window when the table is opened. It is very important to realize that these are just table fields that were copied from the original table. They are not a "handle" into the geometric coordinates that define the position of the points within the drawing's internal structure. Altering the values will not move the points, nor are the fields even necessary to keep the points "positioned" at the right spots.

<table>
<thead>
<tr>
<th>ID</th>
<th>Longitude</th>
<th>Latitude</th>
<th>Name</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>-119.47</td>
<td>39.02</td>
<td>Incine Village</td>
<td>NV</td>
</tr>
<tr>
<td>51</td>
<td>-82.66</td>
<td>27.56</td>
<td>St. Petersburg</td>
<td>FL</td>
</tr>
<tr>
<td>52</td>
<td>-123.42</td>
<td>46.2</td>
<td>Lake Oswego</td>
<td>OR</td>
</tr>
<tr>
<td>53</td>
<td>-76.25</td>
<td>40.17</td>
<td>Lancaster</td>
<td>PA</td>
</tr>
<tr>
<td>54</td>
<td>-77.06</td>
<td>38.91</td>
<td>Alexandria</td>
<td>VA</td>
</tr>
<tr>
<td>55</td>
<td>-74.31</td>
<td>39.49</td>
<td>Margate</td>
<td>NJ</td>
</tr>
<tr>
<td>56</td>
<td>-70.84</td>
<td>42.5</td>
<td>Marblehead</td>
<td>MA</td>
</tr>
<tr>
<td>57</td>
<td>-88.56</td>
<td>44.02</td>
<td>Oshkosh</td>
<td>WI</td>
</tr>
<tr>
<td>58</td>
<td>-104.79</td>
<td>41.15</td>
<td>Cheyenne</td>
<td>WY</td>
</tr>
<tr>
<td>59</td>
<td>-97.67</td>
<td>30.52</td>
<td>Round Rock</td>
<td>TX</td>
</tr>
<tr>
<td>60</td>
<td>-122.11</td>
<td>47.67</td>
<td>Redmond</td>
<td>WA</td>
</tr>
<tr>
<td>61</td>
<td>-97.34</td>
<td>37.59</td>
<td>Wichita</td>
<td>KS</td>
</tr>
<tr>
<td>62</td>
<td>-122.15</td>
<td>37.45</td>
<td>Palo Alto</td>
<td>CA</td>
</tr>
</tbody>
</table>

For example, in the Table window we could delete the "Longitude" field.
We could delete the "Latitude" field as well. If we do so the points do not magically disappear from the drawing. They will still be there. All that we have done is delete a field from the data table associated with the drawing. The points in the drawing have already been created when Manifold read the original database. Changing the text data in the table (either the original one or the new table created with the drawing) changes nothing about those points and does not alter the coordinate numbers inside the drawing.

The above discussion may seem obvious to experienced GIS users; however, for new users it is easy to look at a table full of records with latitude and longitudes and to forget that changing the values in the table does not actually move the points about.

Changing the Location of Points

How then can we change the actual location of points? There are three main ways:

- We can show the Latitude and Longitude intrinsic fields and then change their values. See Editing Intrinsic Fields in Tables. This is the best method when points must be moved to given coordinates.
- We can right click on individual points in the drawing and use the Object - Coordinates dialog to change the location of the geometric coordinates that define the points. This is a seriously tedious approach.
- We can use any one of a wide variety of interactive editing tools in a map or drawing window to move the points by selecting them and dragging them to a new location.

Note that neither of the above options will change the contents of any "latitude" or "longitude" text fields we might see in a table window.

Specifying Languages for Columns

Language specification in text columns varies depending on whether Unicode or ANSI text is used. Unicode text columns can simultaneously contain characters from more than one language. ANSI text columns are usually restricted to one language in addition to (possibly) English. Character data contained in an ANSI text column will be interpreted by the system according to the code page associated with that column.

By default, ANSI text columns use the system code page. If desired, any ANSI text column in a table can be set to use a different code page by right clicking on the column head and choosing Language.
Editing Intrinsic Fields in Tables
We can move objects in drawings by editing intrinsic fields in tables. Before proceeding with this topic, please take a moment to read:

- Intrinsic Fields in Tables
- Editing Data in Tables

Most intrinsic fields are read-only; however, intrinsic fields that report location coordinates may be edited. Changing a coordinate intrinsic field will move the associated object.

The relevant intrinsic fields are:

- **Geom (I)** The geometry data ("the object metric") that defines the position and shape of the object. Read/write. Modifying this column with, say, an UPDATE query, modifies the metric of the associated drawing objects. Setting Geom (I) to a NULL value or to a geometry value the type of which differs from that of the drawing object is not allowed.

- **X (I)** X coordinate of the center of the object in native units.

- **Y (I)** Y coordinate of the center of the object in native units.

- **Latitude (I)** Latitude of the center of the object.

- **Longitude (I)** Longitude of the center of the object.

The X and Y fields report the location of the object using native projection system coordinate units. If a drawing has been projected into some projection other than the Latitude / Longitude projection, these units will be the meter-based units used in that projection. Editing such coordinates is an expert-level task.

Most often we will edit the Latitude and Longitude fields, which report the position of the center of the object in ordinary latitude and longitude degrees. The usual task is to adjust the position of a point to match some specified location given in latitude and longitude degrees. For example, we may have a drawing representing a nautical chart that shows the position of various buoys. Perhaps we have checked the position of those buoys by visiting them and noting their actual location with a GPS and now we wish to adjust the buoy positions in the chart to match our measurements.

There are three main ways of editing intrinsic fields:

- Open the drawing's table, use View - Columns to show the desired intrinsic fields in the table and then edit them like any other field. This is a good method when many items are to be edited, or when other fields in the table (such as, perhaps, a "place name" field) will be used to identify which record is to be edited. This is also a good method when an object will be selected first so that it is obvious which record is to be edited.

- Right click on the object to be moved in the drawing and choose Object Fields context menu. Intrinsic fields will also appear in the Object Fields dialog. This is a good method when objects will be identified by their position.

- Use SQL with an UPDATE query or similar method to edit the Geom (I) intrinsic field. This topic does not cover the use of geoms and instead focuses on the above two methods. See the Geometry in Tables topic for details on working with geoms.

The first two methods are the most interactive for tasks such as adjusting the position of points. Which method is used will largely depend on which provides the easiest identification of the points to be moved. Some drawings may have many thousands of objects and so it is not always easy to find the right record to edit in a table.

**Latitude and Longitude Notation in Editing**

Manifold tables show latitude and longitude intrinsic fields using decimal degrees notation by default. When editing intrinsic fields, Manifold allows entry of values in either decimal or in degrees, minutes, seconds notation using almost any separator between the degrees, minutes and seconds values.
For example, if we want to enter a value of -122.55125 degrees we can double click into the Longitude (I) box and enter any of the following:

-122.55125
-122d33m4.5s
-122 33 4.5
-122°33'4.5"
122.55125W
W122d33m4.5s

Manifold will interpret any non-numeric character in between the numbers other than a decimal point as a separator between degrees, minutes and seconds values. The letters W and S either before or after the other characters are commands to make the value a negative (west) longitude or negative (south) latitude.

Note that case is significant in NESW notation, so that 22d33m4.5s would be a North (positive) latitude and 22d33m4.5S or 22d33m4.5sS would both be South (negative) latitudes.

Whatever notation style is used to enter the new value, it will be displayed using the column format specified for that column.

Moving Areas and Lines

The four coordinate intrinsic fields listed above report the center of the object. In the case of points, the center of the object is the same as the location of the point. Lines and areas, though, are usually defined by many coordinates, so the X, Y, Latitude and Longitude fields for lines and areas will report the centroid location of these objects. This is a considerably more abstract idea than the location of a point.

It's possible to move lines and areas by moving the "center" reported by intrinsic fields. Whatever changes in horizontal or vertical displacement are made will be applied to all of the coordinates defining the object. So, if we moved the center of an area Westward by one degree longitude then all of the coordinates defining the area would be moved Westward one degree.

This capability is less frequently used than moving points by specifying their location via intrinsic fields; however, changing the values of intrinsic fields can be a very handy way of "nudging" areas or lines horizontally or vertically into a desired location.

Objects like areas and lines as well as points can be altered in a mass way by using SQL with an UPDATE query or similar method to edit the Geom (I) intrinsic field. See the Geometry in Tables topic for details on working with geoms.

Native Coordinates

Native coordinates or native units are numbers stored within the drawing that Manifold uses internally as positional coordinates. These numbers are shown in the Object Coordinates dialog that pops up if you right-click an object and choose Coordinates. They are also used in the X (I) and Y (I) intrinsic field columns.

Projected coordinates are native coordinates adjusted with the local scale and local offset parameters of the coordinate system (projection) in use. In many cases, projected coordinates are the same as native coordinates (because of the local scale and offset values) but sometimes they are different.

Advanced Uses

Intrinsic fields can be used like other fields in a table. For example, we could compute altered latitude and longitude coordinates via an Active Column and then copy the Active Column into the Latitude and Longitude fields using the transform toolbar.

Enterprise Edition
When working with Enterprise Edition, editing an intrinsic field for a shared drawing will not be possible if the drawing and drawing table are both read-only. Check out the drawing and the table to edit intrinsic fields.

**See Also**

See the Add Points with Instant Data example topic for an example of moving points by editing intrinsic fields.

See the Geometry in Tables topic for details on working with geoms.
Regular Expressions

Regular expressions are a concise and flexible notation for finding and replacing patterns with the Edit - Find, Edit - Find Next and Edit - Replace dialogs. There are several different syntax styles used for regular expressions in computing. Manifold uses the regular expression syntax used in the Microsoft .NET framework, the latest edition of the Microsoft regular expression syntax familiar to many from Microsoft ActiveX scripting languages. Note that .NET introduces slight changes from the regular expression syntax used in both Jscript and VBScript. Microsoft regular expression syntax is very similar to that used in UNIX/Linux grep command. ("grep" = "global regular expression processor").

Special characters and sequences are used in writing patterns for regular expressions. The following table describes these characters and includes short examples showing how the characters are used.

<table>
<thead>
<tr>
<th>Character</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\</td>
<td>Marks the next character as either a special character or a literal. For example, &quot;n&quot; matches the character &quot;n&quot;, &quot;\n&quot; matches a newline character. The sequence &quot;\l&quot; matches &quot;\l&quot; and &quot;\t&quot; matches &quot;\t&quot;.</td>
</tr>
<tr>
<td>^</td>
<td>Matches the beginning of input.</td>
</tr>
<tr>
<td>$</td>
<td>Matches the end of input.</td>
</tr>
<tr>
<td>.</td>
<td>Matches any single character except a newline character.</td>
</tr>
<tr>
<td>(pattern)</td>
<td>Matches pattern and remembers the match. The matched substring can be retrieved from the resulting Matches collection, using Item [0]...[n]. To match parentheses characters ( ), use &quot;(&quot; or &quot;)&quot;.</td>
</tr>
<tr>
<td>x</td>
<td>y</td>
</tr>
<tr>
<td>n</td>
<td>is a nonnegative integer. Matches exactly n times. For example, &quot;o{2}&quot; does not match the &quot;o&quot; in &quot;Bob,&quot; but matches the first two o's in &quot;fooood&quot;.</td>
</tr>
<tr>
<td>{n}</td>
<td>n is a nonnegative integer. Matches at least n times. For example, &quot;o{2,}&quot; does not match the &quot;o&quot; in &quot;Bob&quot; and matches all the o's in &quot;foooood&quot;. &quot;o{1,}&quot; is equivalent to &quot;o+&quot;, &quot;o{0,}&quot; is equivalent to &quot;o*&quot;.</td>
</tr>
<tr>
<td>{n,m}</td>
<td>m and n are nonnegative integers. Matches at least n and at most m times. For example, &quot;o{1,3}&quot; matches the first three o's in &quot;fooood&quot;. &quot;o{0,1}&quot; is equivalent to &quot;o?&quot;.</td>
</tr>
<tr>
<td>[xyz]</td>
<td>A character set. Matches any one of the enclosed characters. For example, &quot;[abc]&quot; matches the &quot;a&quot; in &quot;plain&quot;.</td>
</tr>
<tr>
<td>[^xyz]</td>
<td>A negative character set. Matches any character not enclosed. For example, &quot;[^abc]&quot; matches the &quot;p&quot; in &quot;plain&quot;.</td>
</tr>
<tr>
<td>[a-z]</td>
<td>A range of characters. Matches any character in the specified range. For example, &quot;[a-z]&quot; matches any lowercase alphabetic character in the range &quot;a&quot; through &quot;z&quot;.</td>
</tr>
<tr>
<td>[^m-z]</td>
<td>A negative range characters. Matches any character not in the specified range. For example, &quot;[^m-z]&quot; matches any character not in the range &quot;m&quot; through &quot;z&quot;.</td>
</tr>
<tr>
<td>\b</td>
<td>Matches a word boundary, that is, the position between a word and a space. For example, &quot;er\d&quot; matches the &quot;er&quot; in &quot;never&quot;</td>
</tr>
</tbody>
</table>
but not the "er" in "verb".

\B Matches a nonword boundary. "ea*ri\B" matches the "ear" in "never early".

\d Matches a digit character. Equivalent to [0-9].

\D Matches a nondigit character. Equivalent to [^0-9].

\f Matches a form-feed character.

\n Matches a newline character.

\r Matches a carriage return character.

\s Matches any white space including space, tab, form-feed, etc. Equivalent to [ \f\n\r\t\v].

\S Matches any nonwhite space character. Equivalent to [^ \f\n\r\t\v].

\t Matches a tab character.

\v Matches a vertical tab character.

\w Matches any word character including underscore. Equivalent to [A-Za-z0-9_].

\W Matches any nonword character. Equivalent to [^A-Za-z0-9_].

\num Matches \num, where \num is a positive integer. A reference back to remembered matches. For example, "(\.)\1" matches two consecutive identical characters.

\n Matches \n, where \n is an octal escape value. Octal escape values must be 1, 2, or 3 digits long. For example, "\111" and "\011" both match a tab character. "\0011" is the equivalent of "\001" & "1". Octal escape values must not exceed 256. If they do, only the first two digits comprise the expression. Allows ASCII codes to be used in regular expressions.

\n Matches \n, where \n is a hexadecimal escape value. Hexadecimal escape values must be exactly two digits long. For example, "\x41" matches "A". "\x041" is equivalent to "\x04" & "1". Allows ASCII codes to be used in regular expressions.

For details on options in the Find and Replace dialogs, see the Edit - Find / Find Next and Edit - Replace topics.

Regular Expressions in Find and Replace

Regular expressions can be used in the Find box to find items and also in the Replace box to specify how they should be replaced. For example, Europeans will often denote decimal places with a comma instead of a period as is used in the US. This transformation can be easily done with a Replace command using regular expressions.

Strings like "101.999" can be transformed to "101,909" (comma instead of period) by supplying a Find string of \([0-9]{3}\).\([0-9]{3}\) and a Replace string of $1, $2.
Note that the above is a search and replace on a *string* (that is, a text) value. Changing the period or comma notation in numeric values is done in the Windows localization settings.

### Regular Expression Examples

The following examples list a regular expression in **bold face** that can be used to search for various patterns. Each regular expression is then followed by sample strings and whether or not the string would be OK in that regular expression or if it would fail the test posed by the regular expression.

These examples show how regular expressions can be used to find certain patterns. They do not purport to be definitive filters for the various examples given. For example, a rigorous filter for URLs would be a more complex regular expression than that that provided below since it would exclude characters such as # that are not allowed in domain names. The last example, for email addresses, shows a more robust regular expression that can be used as a true validation filter for acceptable email addresses.

**First name:** `{Carlos|Mario}`s.*

- Carlos Cramer: OK
- Carlos Hernandez: OK
- Carlos Gonzalez: OK
- Mario Hernandez: OK
- Paolo Accorti: Fail

**Last name:** `\S+\s+Hernandez`

- Carlos Cramer: Fail
- Carlos Hernandez: OK
- Carlos Gonzalez: Fail
- Mario Hernandez: OK
- Paolo Accorti: Fail

**First and last name:** `Carlos's(?;Hernandez|Cramer)`

- Carlos Cramer: OK
<table>
<thead>
<tr>
<th>Name</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carlos Hernandez</td>
<td>OK</td>
</tr>
<tr>
<td>Carlos Gonzalez</td>
<td>Fail</td>
</tr>
<tr>
<td>Mario Hernandez</td>
<td>Fail</td>
</tr>
<tr>
<td>Paolo Accorti</td>
<td>Fail</td>
</tr>
</tbody>
</table>

US style phone number: `(\+\d)?(\(\d+\))?\d\([\d-]*`

- `+7(514)555-9931` OK
- `(514) 333-9931` OK
- `(617) 555-3267` OK
- `555-8787` OK
- `(1) 03.83.00.68` Fail

Phone number containing area code 514: `(\+\d)?\(514\)\d\([\d-]*`

- `+7(514)555-9931` OK
- `(514) 333-9931` OK
- `(617) 555-3267` Fail
- `555-8787` Fail
- `(1) 03.83.00.68` Fail

Phone number starting with 555: `(\+\d)?\d*\(\(\d+\)\)\d*\d*`

- `+7(514)555-9931` OK
- `(514) 333-9931` Fail
- `(617) 555-3267` OK
- `555-8787` OK
- `(1) 03.83.00.68` Fail

URL: `(?:ftp:\\|http:\\|mailto:\\)?(\w+@)?(www\.)?\w+\(\(\w+\)+\(\:\d+)?)`

- `http:\www.manifold.net:8080` OK
- `http:\manifold` Fail
- `www.manifold.net` OK
- `ftp:\\microsoft.com` OK
- `mailto:\john@manifold.com` OK
Manifold URL: (?ftp:\\|http:\\|mailto:\\)?(\w+@)?(www\.)?manifold(\.|\w+)+(:\d+)?

- http://www.manifold.net:8080 OK
- http://manifold Fail
- www.manifold.net OK
- ftp:\\microsoft.com Fail
- mailto:\\john@manifold.com OK

FTP URL: ftp:\\(www\.)?\w+(\.|\w+)+(:\d+)?

- http://www.manifold.net:8080 Fail
- http://manifold Fail
- www.manifold.net Fail
- ftp:\\microsoft.com OK
- mailto:\\john@manifold.com Fail

Latitude: \d+°\d+\'\d+(\.|\d+)?"\s*(N|S)?

- 0°00'00.00" OK
- 83°02'50.82" N OK
- 0°05'43.14" S OK
- 9' Fail

South latitude: \d+°\d+\'\d+(\.|\d+)?"\s*S

- 0°00'00.00" Fail
- 83°02'50.82" N Fail
- 0°05'43.14" S OK
- 9' Fail

Date: \d+(\/|-)\d+(\/|-)\d+

- 5/24/1985 OK
- 5/24/85 OK
- 5-24-1985 OK
- 5.24.1985 Fail
Tables

<table>
<thead>
<tr>
<th>Date Format</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>5/24/1985</td>
<td>Fail</td>
</tr>
<tr>
<td>5-24-1995</td>
<td>OK</td>
</tr>
<tr>
<td>5.24.1995</td>
<td>Fail</td>
</tr>
</tbody>
</table>

Seeking a date of the 24th:

<table>
<thead>
<tr>
<th>Date Format</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>5/24/1985</td>
<td>OK</td>
</tr>
<tr>
<td>5-29-1985</td>
<td>Fail</td>
</tr>
<tr>
<td>5.24.1985</td>
<td>Fail</td>
</tr>
</tbody>
</table>

Dollar currency: \(\$.*\d+(\./\d+)?\)|\(\d+(\./\d+)?\)\$\)

<table>
<thead>
<tr>
<th>Value</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>$5</td>
<td>OK</td>
</tr>
<tr>
<td>4.6 $</td>
<td>OK</td>
</tr>
<tr>
<td>-7.3 $</td>
<td>Fail</td>
</tr>
<tr>
<td>7.3</td>
<td>Fail</td>
</tr>
</tbody>
</table>

Exponential number: \(-?\d+\d+[E|D](-?\d+)\)?

<table>
<thead>
<tr>
<th>Value</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>-3.5E2</td>
<td>OK</td>
</tr>
<tr>
<td>3D-56</td>
<td>OK</td>
</tr>
<tr>
<td>-7</td>
<td>OK</td>
</tr>
<tr>
<td>.8</td>
<td>Fail</td>
</tr>
</tbody>
</table>

Rational fraction: \(-?((\d+\d+)?\d+)/(\d+))\)

<table>
<thead>
<tr>
<th>Value</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>-2</td>
<td>OK</td>
</tr>
<tr>
<td>2/5</td>
<td>OK</td>
</tr>
<tr>
<td>-1 2/5</td>
<td>OK</td>
</tr>
<tr>
<td>2/-5</td>
<td>Fail</td>
</tr>
</tbody>
</table>

HTML tag: <\(.*\)>.*</\1>

<table>
<thead>
<tr>
<th>Tag</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;a&gt;abc&lt;/a&gt;</td>
<td>OK</td>
</tr>
<tr>
<td>&lt;a&gt;abc&lt;/b&gt;</td>
<td>Fail</td>
</tr>
<tr>
<td>&lt;a&gt;abc</td>
<td>Fail</td>
</tr>
</tbody>
</table>
SSN (social security number): \d{3}-\d{2}-\d{4}

- 223-20-9898 OK
- 22-20-9898 Fail
- 223-209898 Fail
- 223 20 9898 Fail

Credit card style number patterns: \d{4}( \d{4}){3,4}

- 2235 5656 4578 7890 OK
- 2235 5656 4578 7890 0010 OK
- 2235 5656 4578 7890 00 Fail
- 2235-5656-4578-7890 Fail

Email address validation: ([\w.\!#$%\&\'\(\)\*\+,\-\./\0-9;\=\?\@\[\]\^\`\{\}\|\-\+]+@[A-Za-z0-9-]+\.[A-Za-z0-9-]+)+)

- john_smith@domain.com OK
- john.smith@domain.com.au OK
- john_smith.domain.com.au Fail
- john smith@domain.com.au Fail
- john_smith@domain Fail
- john_smith@ Fail
Adding or Deleting Fields in Tables
Deleting a field from a table is easy: right click on the column header for the field to be deleted and choose Delete from the context menu.

Add a field by right clicking on any column head and choosing Add from the context menu. We can also add a field through the Table - Design dialog.
Changing Field Types in Tables
Change field types in tables between compatible types by using the Transform toolbar.

To change field types:

1. Open the table that contains the target records.
2. Right click onto the column head for the field that is to be changed and choose Change Type from the context menu.

Note that field type conversions are only possible between compatible types. Conversions are also subject to the limitations of their types. For example, any number can be converted into a text string but not every text string can be converted into a number.

Conversions between text strings and numbers will follow Microsoft conventions. For example, a text string such as "(254)" will be converted into -254, a negative number, since numbers in parentheses are normally used in accounting tables to indicate negative numbers.

Conversions can also alter data. For example, converting from a floating point number or a text string such as "3.1415" to an integer will round down to an integer value of 3 while copying a floating point number or a text string such as "3.6" will round up to 4. Rounding occurs, of course, because integers cannot show fractional values.

If Manifold's conversion routines cannot make sense of a requested conversion then no conversion will occur.

Example

We have a table with a field called Area Codes that is a text field. We wish to convert it to a numeric field. We first create a new integer field called temp in the Design dialog. Next, we use the transform tool bar loaded with:

Area Codes  Copy to  temp

Press Apply to copy the contents of the Area Codes column into the temp column. The transform Copy to operation will automatically translate field types on the fly between compatible types, so that text strings of the form "702" will be converted to the integer number 702.

We can now delete the original Area Codes column and rename the temp column to be called Area Codes.

Transform Toolbar - Tables

<table>
<thead>
<tr>
<th>Target box</th>
<th>Operation box</th>
<th>Source / Argument box</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discount</td>
<td>Acid</td>
<td>.05</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Apply</td>
</tr>
</tbody>
</table>

The Transform toolbar makes changes to an entire column (field) throughout the table. If a selection is present it will operate only on those records in the selection. If no selection is present it will apply to all records in the table.

The Transform toolbar consists of three boxes, from left to right: a target box, the operation box, and a source / argument box.

Target box Also known as the scope box. The column that will be affected, altered or which will control the operation. The target box will be pre-loaded with the names of all fields in the table. The illustration above shows the Discount field has been chosen.
### Operation box

The function to be applied. The operation box is context sensitive and will show only those operations that make sense for the field type of the target column. For example, arithmetic operations such as **Add** will only be enabled for numeric fields.

### Source / Argument box

The value to be used. Depending on the operator, this may be another column or a value entered by the user. A few operators (such as **Clear**) do not require a source or argument. The source / argument box will not be enabled for such operators. The illustration above shows we have entered `.05` as a value in the source / argument box.

We use **dark blue**, **black**, and **violet** bold face fonts in this topic to distinguish the three boxes to make examples and explanation more clear. In real life Manifold uses the same black font color in all three Transform toolbar boxes.

### Using the Transform toolbar

1. Open the table that contains the target records.
2. Make a selection if the operation is to be applied just to the selection.
3. Choose the target column in the target box.
4. Choose the desired operator in the operation box.
5. Choose or specify a value in the source / argument box, if this operation requires it.
6. Press **Apply**.

### Target box

The left-most box specifies the target field. This is the column that will be affected, altered or which will control the operation. Choosing the name of a column will apply the Transform function to only that field. The example above shows we will apply the Transform function to the **Discount** column in the table. If a selection is present then whatever operation we apply with the Transform toolbar will act only on those records that are part of the current selection. If no selection is present the transform operation will apply to all records in the table.

Note that the function of the leftmost box is slightly different when the Transform toolbar is used with tables as compared to images or drawings. The target box with tables is used to specify the name of a column to which the function should be applied. The scope (that is, all records or only the selection) is automatic based on whether there is a selection present or not.

With images and drawings the leftmost box is known as a **target box** and is just an information box. It is automatically loaded with the scope (all objects/pixels, or the selection, if one is present) based on whether there is a selection present or not.

### Operation box

Choose a function from the long list of operators available in the operation box. The example shows **Add**, which is enabled for numeric fields and which adds the value in the source / argument box to all values in the target column. The Transform Operators - Tables topic lists operators available for use with tables.

### Source / argument box

`.05`
Some operators do not require any parameters. For these functions the source / argument box will be disabled. Other functions will require specification of an additional parameter. This may be another column (such as the choices in the target box), or it could be a parameter that is entered by the user as illustrated above. To enter a parameter, click into the source / argument box, enter the value using the keyboard and then press Enter.

Examples

[Discount] Add .05 - Add .05 to the value in the Discount field. (Sounds like we're giving everyone an additional 5% off!)

[Elevation] Multiply by .3048 - Multiply all values in the Elevation field by .3048 (Clearly, we are converting a database of elevations given in feet into meters).

[Elevation] Multiply by [Offset] - For each record, take the value of the Elevation field in that record and multiply it by the value of the Offset field in that record and put the result into the Elevation field.

Type Conversions and Compatibility

There are four field type families in Manifold tables for transform toolbar purposes:

- **Date** - Date and time field type.
- **Numeric** - Currency, integer, floating-point, latitude and longitude field types.
- **Percentage** - Percentage field type.
- **Text** - Text and URL field types.

Manifold will automatically convert types on the fly when copying from one data type to a compatible data type using transform operators. Within the same field type family most operations will be performed automatically with few issues other than the most obvious: floating point numbers will be rounded when converted into integers.

When using transform operators between fields of different types not all conversions are possible, since not all data can exist in compatible forms for all data types. For example a text field can contain text values like "123" and "1984" that can be converted into numbers; however, spelled out numbers like "one two three" and mixed values like "43-B5" cannot be converted into numbers. If we copy values like "43-B5" from a text column into an integer column the result will be zero. Manifold will do the best job possible given standard type conversion conventions as applied within Windows Visual C++ and Visual Basic programming.

A few operators don’t make sense unless the field types in the target box and the source / argument box are in the same field type family. These are the homogeneous statistic functions (impress your friends with that nomenclature!): Fill with Maximum, Fill with Minimum, Fill with Median, Make at Least and Make at Most. For these transform operators the target box and the source / argument box fields must be of the same field type. For example, one cannot take a text field and use Make at Least to make it at least like the contents of an Integer field. As a theoretical matter, Manifold could have been internally arranged to make conversions on the fly in a way that would have allowed these functions to have been applied in one step between incompatible field type families. As a practical matter doing so would have involved numerous operational subtleties that are very difficult for most people to understand on the fly.

Instead, such functions are applied between different field type families using a two step process: The correct way to apply a homogeneous statistic function between fields from different field type families is to first make a conversion using a copy and then to apply the statistic function. For example, to apply Make at Least to a text field based on an integer field, first apply Make at Least between the integer field into another integer field and then copy the value of that integer field into the target text field.
Transform Operators - Tables

When used with tables, the Transform Toolbar provides numerous functions for use with fields of various types.

The leftmost box in the Transform toolbar is the Target box. The rightmost box is the Source / argument box. Some operators require choice of a field in the source / argument box while other operators allow either a value entered by the user or choice of a field.

The list of operators in the operation box will vary depending on the type of field chosen in the Target box. For example, Round, Round Up, and Round Down will be listed in available operators only with float or double types and not with integers. Conversely, the Binary And and similar operators will not be listed with float types because these are bitwise operators that make sense only with integers.

When working with tables, it is at times important to know what data type each field is. For example, the value 486 could either be an integer type or it could be a text string consisting of the characters "4", "8" and "6". When working with a column that contains such values, if the operators you expect to see are not there remember to double-check the field type in case it is different than expected.

Some operators that appear to be numeric in character will appear with text fields. Such operators will place their resulting numbers into the text field as a string.

With text fields, "values" mean text strings or characters. All characters are taken literally. "Maximum" and "minimum" functions employ lexicographic arithmetic taking the order of characters in the character set as the numeric value of the character.

Tokens are explained in the Transform - Using Tokens and Text Strings topic.

Date / Time Decrement and Increment Operations

Manifold System date-time operations will automatically "rollover" when incrementing or decrementing. For example, if the Inc Minute command is applied to a time of 2:59 the result will be 3:00. Applying the Dec Minute command to 3:00 will result in 2:59.

Impossible Operations Ignored

Manifold has a simple way of dealing with impossible operations such as division by zero. All such operations are ignored and any records that would be subject to such operations are left unchanged. It is therefore important when using operations such as Divide by to first check the parameter involved (say, by sorting on that column) to see if there are any zero values.

Span Excluding / Including Operators

See the topic Transform - Span Excluding / Including for discussion and examples.

Date/Time and URL Fields have Subparts Called "Sections"

A Date/Time field may be subdivided into subparts such as, for example, the hour, minutes and seconds value of the time. These subparts are called sections. Likewise, the date part of a Date/Time field has sections such as the year, month and day. URL fields also have sections such as the object, parameters, port, protocol and server.

Any operator that works on date/time or URL field sections will apply only to that section. So, for example, using Copy Year from with Date time fields in both the target box and the source / argument box will copy only the year from the parameter date field to the target box's year section.

The only two exceptions to the above are the Copy Month Name From and Copy Weekday Name From operators, which can only copy from a date time field section to a text field or a URL section. The reason is that these operators convert date/time data into a name, such as "Tuesday", which is a one-way conversion. Any particular date can unambiguously be converted into a day of the week; however, simply knowing a day is "Tuesday" does not pick out unambiguously a calendar date.
**Text Operators**

- **Append** Append given text or field to original field.
- **Clear** Clear all values in Field.
- **Copy Day Name from** Examine the date/time in the Parameter field and copy the name of the day to Field using local language settings.
- **Copy First Token from** Copy the first token from Parameter into the Field. If Parameter contains John/Paul/George/Ringo the result would be to copy John into Field.
- **Copy from** Copy the value from the Parameter into the Field.
- **Copy Last Token from** Copy the last token from Parameter into the Field. If Parameter contains John/Paul/George/Ringo the result would be to copy Ringo into Field.
- **Copy Month Name from** Examine the date/time in the Parameter field and copy the name of the month to Field using local language settings.
- **Copy URL Object from** Copy main part of URL field.
- **Copy URL Parameters from** Copy query parameters, if any, in URL.
  - Example: "\texttt{?pq=q\&sc=on}\" from http://www.mydomain.com/cgi-bin/query?pg=q&sc=on
- **Copy URL Port from** Return port, if used, from URL string.
- **Copy URL Protocol from** Return protocol used by the URL.
  - Example: "\texttt{http://}\" from http://www.manifold.net
- **Copy URL Server from** Return the server from the URL: Eg: "\texttt{www.manifold.net}\" from http://www.manifold.net/news/news.html
- **Decode with** Decrypt Field using a password given in Parameter.
- **Delete Containing** Clear values in original field that contain the given text or field.
- **Delete Containing Match** Clear all values that contain a match to the given regular expression pattern.
- **Delete Containing Token** Clear all values that contain the given token.
- **Delete Ending with** Clear values in original field that end with the given text or field.
- **Delete Ending with Match** Clear all values that end with a match to the given regular expression pattern.
- **Delete Ending with Token** Clear all values that end with the given token.
- **Delete First Token** Delete the first token in the Field together with the separator (if any) following it. If applied to John/Paul/George/Ringo the result would be Paul/George/Ringo.
<table>
<thead>
<tr>
<th><strong>Delete Last Token</strong></th>
<th>Delete the last token in the Field together with the separator (if any) preceding it. If applied to John/Paul/George/Ringo the result would be John/Paul/George.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Delete Left</strong></td>
<td>Delete the first n characters of the original field, where n is a number specified by the given numeric field or value.</td>
</tr>
<tr>
<td><strong>Delete Matching</strong></td>
<td>Clear all values that match the given regular expression pattern.</td>
</tr>
<tr>
<td><strong>Delete Right</strong></td>
<td>Delete the last n characters of the original field, where n is a number specified by the given numeric field or value.</td>
</tr>
<tr>
<td><strong>Delete Sounding Like</strong></td>
<td>Clear all values that sound like the given word. Requires a lot of faith in the Soundex algorithm. A better idea is to use a viewbot to select all that sound like the given word and then take a look at the selection before deleting.</td>
</tr>
<tr>
<td><strong>Delete Starting with</strong></td>
<td>Clear values in original field that start with the given text or field.</td>
</tr>
<tr>
<td><strong>Delete Starting with Match</strong></td>
<td>Clear all values that start with a match to the given regular expression pattern.</td>
</tr>
<tr>
<td><strong>Delete Starting with Token</strong></td>
<td>Clear all values that start with the given token.</td>
</tr>
<tr>
<td><strong>Encode with</strong></td>
<td>Encrypt Field using a password given in Parameter. Caution: don't &quot;Encode&quot; a field twice by accident!</td>
</tr>
<tr>
<td><strong>Fill with</strong></td>
<td>Fill the Fields of all records with what is in the Parameter. Equivalent to &quot;Copy from&quot; the Parameter value or field.</td>
</tr>
<tr>
<td><strong>Fill with Maximum of</strong></td>
<td>Find the &quot;maximum&quot; value that occurs in Parameter field using lexicographic arithmetic and copy it to the Field.</td>
</tr>
<tr>
<td><strong>Fill with Median of</strong></td>
<td>Find the &quot;median&quot; value that occurs in Parameter field using lexicographic arithmetic and copy it to the Field.</td>
</tr>
<tr>
<td><strong>Fill with Minimum of</strong></td>
<td>Find the &quot;minimum&quot; value that occurs in Parameter field using lexicographic arithmetic and copy it to the Field.</td>
</tr>
<tr>
<td><strong>Leave Containing</strong></td>
<td>Clear all values in original field that do not contain the given text or field.</td>
</tr>
<tr>
<td><strong>Leave Containing Match</strong></td>
<td>Clear all values except those that contain a match to the given regular expression pattern.</td>
</tr>
<tr>
<td><strong>Leave Containing Token</strong></td>
<td>Clear all values except those that contain the given token.</td>
</tr>
<tr>
<td><strong>Leave Ending with</strong></td>
<td>Clear all values in original field that do not end with the given text or field.</td>
</tr>
<tr>
<td><strong>Leave Ending with Match</strong></td>
<td>Clear all values except those that end with a match to the given regular expression pattern.</td>
</tr>
<tr>
<td><strong>Leave Ending with Token</strong></td>
<td>Clear all values except those that end with the given token.</td>
</tr>
<tr>
<td><strong>Leave Left</strong></td>
<td>Clip all text from the Field except the first n characters, where n is a number specified by the value given in the Source / argument box.</td>
</tr>
<tr>
<td><strong>Leave Matching</strong></td>
<td>Clear all values except those that match the given regular expression pattern.</td>
</tr>
</tbody>
</table>
Leave Right  Clip all text from the Field except the last \( n \) characters, where \( n \) is a number specified by the value given in the Source / argument box.

Leave Sounding Like  Clear all values except those that sound like the given word. Not for the faint of heart. A better idea is to use a viewbot to select all that sound like the given word and then take a look at the selection, invert the selection and then delete.

Leave Starting with  Clear all values in original field that do not start with the given text or field.

Leave Starting with Match  Clear all values except those that start with a match to the given regular expression pattern.

Leave Starting with Token  Clear all values except those that start with the given token.

Make Lower Case  Convert all characters in the Field to lower case.

Make Sentence Case  Capitalize first character in sentences. Example: "small green tree" becomes "Small green tree"

Make Title Case  Capitalize first character in words. Example: "small green tree" becomes "Small Green Tree"

Make Upper Case  Convert all characters in the Field to upper case.

Prepend  Prepends given text or field to original field.

Reverse  Reverses the order of characters in the Field.

Reverse by Tokens  Reverses the order of tokens in the Field. If applied to John/Paul/George/Ringo the result would be Ringo/George/Paul/John.

Reverse Token at  Take the token in the Field at the position given by the number in the value box and reverse its characters. If applied to John/Paul/George/Ringo with a value of 3 would result in John/Paul/egroeG/Ringo.

Span Excluding  Start at the first character in the Field and see if it occurs anywhere in the text string in the Source / argument box. If it does not occur, continue to the next character and see if it occurs in the text string in the Source / argument box. Continue until a character is encountered that is in the text string in the Source / argument box and stop. Take the characters checked thus far in order and place them in the Field. Order in the Field is preserved, while the order of characters in the text string in the Source / argument box or field does not matter.

Span Including  Start at the first character in the Field and see if it occurs anywhere in the text string in the Source / argument box or field. If it does occur, continue to the next character and see if it occurs in the text string in the Source / argument box and stop. Take the characters found thus far in order and place them in the Field. Order in the Field is preserved, while the order of characters in the text string in the Source / argument box or field does not matter.

Toggle Case  Swap upper case for lower case throughout the text.

Trim  Remove leading and trailing separator characters (whitespace, by default) from the Field.

Trim Left  Remove leading separator characters (whitespace, by default) from the Field.
The **Span Including** and **Span Excluding** operators are easier to use than their technical description makes it seem. Both chop up a string at a given cut point and return all characters encountered before the cut point. The difference is in how the operators specify the cut point.

The **Including** version cuts off the string as soon as it finds a character **not** in the set of characters given in the Source / argument box.

The **Excluding** version cuts off the string as soon as it finds a character in the set of characters given in the Source / argument box.

For example, suppose we want to cut a string at the first non-numeric character. We would use **Span Including** and provide the characters `1234567890` in the Source / argument box and each string would be cut as soon as a non-numeric character is encountered. For example, the string `123 Main St` would be cut right after the `123` and the operator would return the value `123`.

Suppose we want to cut a string at the first numeric character encountered. We would use **Span Excluding** and again provide the characters `1234567890` in the Source / argument box and each string would be cut as soon as a numeric character is encountered. For example, the string `Cambridge, MA 02138` would be cut right before the `0` character and the operator would return the value `Cambridge, MA` complete with trailing space character.

There are two **Span** operators because sometimes it is more convenient to specify what you are looking for and sometimes it is more convenient to specify what you do not want. For example, there are fewer numeric characters than non-numerics so if you are looking for non-numerics it is easier to supply a short list of numbers than a very long list of all other characters.

The **Trim** operators remove characters listed in the Tool Properties pane’s list of token separators. By default, these are the “white space” characters consisting of the space character, tab, newline and carriage return. Note that adding any other characters to the separator list in Tool Properties will subject them to removal as well if they occur as leading or trailing characters.

### Integer Operators

- **Add**
  Add Parameter field or value to the Field for each record.

- **Arithmetic Series**
  Fill Field with an arithmetic series from first record to last computed using the value **n** given in the Source / argument box for each step. If **n** = 5, Field will contain 0, 5, 10, 15, 20... series values in records from top to bottom.

- **Binary And with**
  Performs bitwise AND operation on Field and given Parameter value or field, and stores the result in the Field.

- **Binary Not**
  Performs bitwise NOT operation on Field. This is a binary inversion: all 0’s become 1’s and all 1’s become 0’s.

- **Binary Or with**
  Performs bitwise OR operation on Field and given Parameter value or field.

- **Binary Xor with**
  Performs bitwise XOR operation on Field and given Parameter value or field.

- **Clear**
  Delete all values in Field.

- **Copy Day from**
  Examine the date/time in the Parameter field and copy the day number (as the day number in the month) to Field.

- **Copy Day of Week from**
  Examine the date/time in the Parameter field and copy the day number (as the day number in the week) to
Copy Day of Year from
Examine the date/time in the Parameter field and copy the day number (as the day number in the year) to Field.

Copy from
Copy values from the given Parameter field to Field.

Copy Hour from
Examine the date/time in the Parameter field and copy the hour number (as the hour number in a 24 day) to Field.

Copy Leap Year Flag from
Examine a date/time field in the Parameter field, determines whether or not each date value falls on a leap year, and copies the result (0 or 1) into Field, which should be a numeric or Boolean field.

Copy Minute from
Examine the date/time in the Parameter field and copy the minute number (as the minute number in an hour) to Field.

Copy Month from
Examine the date/time in the Parameter field and copy the month number (as the month number in a year) to Field.

Copy Percentage from
Sum values in Parameter field and then for each record take the Parameter field’s value as a percentage of the total and copy into Field.

Copy Record Deviation from
Find the median of all Parameter values and then for each record place in Field the deviation from that median for the particular Parameter value of that record.

Copy Record Variation from
Find the median of all Parameter values and then for each record place in Field the variation from that median for the particular Parameter value of that record.

Copy Running Product from
For each record in the sort order, form a product of Parameter field with the cumulative product of multiplying Parameter field from previous records. Place the result in the current record’s Field.

Copy Running Sub from
For each record in the sort order, subtract Parameter field from the cumulative total of subtracting Parameter field from previous records. Place the result in the current record’s Field.

Copy Running Total from
For each record in the sort order, add Parameter field to the cumulative total of subtracting Parameter field from previous records. Place the result in the current record’s Field.

Copy Second from
Examine the date/time in the Parameter field and copy the second number (as the second number in a minute) to Field.

Copy Week from
Examine the date/time in the Parameter field and copy the week number (as the week number in a 52-week year) to Field.

Copy Year from
Examine the date/time in the Parameter field and copy the year number to Field.

Divide by
Divides Field by the given Parameter value or field. Division by zero results in no change to Field for those records.

Fill with
Fill the Fields of all records with what is in the Parameter. Equivalent to "Copy from" the Parameter value or field.

Fill with Average of
Examine Parameter field values and copy average of
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fill with Excess of</td>
<td>Examine Parameter field values and copy excess of these to Field for all records.</td>
</tr>
<tr>
<td>Fill with Maximum of</td>
<td>Examine Parameter field values and copy maximum of these to Field for all records.</td>
</tr>
<tr>
<td>Fill with Median of</td>
<td>Examine Parameter field values and copy median of these to Field for all records.</td>
</tr>
<tr>
<td>Fill with Minimum of</td>
<td>Examine Parameter field values and copy minimum of these to Field for all records.</td>
</tr>
<tr>
<td>Fill with Random</td>
<td>Fill Field with random numbers.</td>
</tr>
<tr>
<td>Fill with Skewness of</td>
<td>Examine Parameter field values and copy skewness of these to Field for all records.</td>
</tr>
<tr>
<td>Fill with StdDeviation of</td>
<td>Examine Parameter field values and copy standard deviation of these to Field for all records.</td>
</tr>
<tr>
<td>Fill with Sum of</td>
<td>Examine Parameter field values and copy sum of these to Field for all records.</td>
</tr>
<tr>
<td>Fill with Variance of</td>
<td>Examine Parameter field values and copy variance of these to Field for all records.</td>
</tr>
<tr>
<td>Geometric Series</td>
<td>Fill Field with a geometric series from first record to last computed using the value n given in the Source / argument box for each step. If n = 2, Field will contain 1, 2, 4, 8, 16, 32 ...series values in records from top to bottom.</td>
</tr>
<tr>
<td>Logical And with</td>
<td>Places a 0 or a 1 in the Field based on logical AND with the field or value specified in the Source / argument box. Any nonzero value counts as a 1 for the AND operation.</td>
</tr>
<tr>
<td>Logical Not</td>
<td>If the Field contains 0, it is made a 1. If it contains any nonzero value it is made a 0.</td>
</tr>
<tr>
<td>Logical Or with</td>
<td>Places a 0 or a 1 in the Field based on logical OR with the field or value specified in the Source / argument box. Any nonzero value counts as a 1 for the OR operation.</td>
</tr>
<tr>
<td>Logical Xor with</td>
<td>Places a 0 or a 1 in the Field based on logical XOR with the field or value specified in the Source / argument box. Any nonzero value counts as a 1 for the XOR operation.</td>
</tr>
<tr>
<td>Make Absolute</td>
<td>Take absolute value of the Field and place it in the Field.</td>
</tr>
<tr>
<td>Make at Least</td>
<td>Resets all values in the Field which are lower than the field or value specified in the Parameter box to the field or value specified in the Parameter box. A lower bound clipping function controlled by the field or value specified in the Parameter box.</td>
</tr>
<tr>
<td>Make at Most</td>
<td>Resets all values in the Field which are higher than the field or value specified in the Parameter box to the field or value specified in the Parameter box. An upper bound clipping function controlled by the field or value specified in the Parameter box.</td>
</tr>
<tr>
<td>Modulo</td>
<td>Replace the Field with Field modulo value or field given in the Source / argument box.</td>
</tr>
<tr>
<td>Multiply by</td>
<td>Multiplies Field by given field or value in Source / argument box.</td>
</tr>
</tbody>
</table>
Negate  Multiply Field by -1 (switches sign).

Power to  Raise Field value to the power specified by the given Parameter value or field.

Square  Replace Field with square of Field.

Square Root  Replace Field with square root of Field.

Subtract  Subtracts the field or value specified in the Source / argument box from the Field.

Floating Point Operators

Add  Add Parameter field or value to the Field for each record.

Arc Cosine  Replace Field with arc cosine of Field.

Arc CoTangent  Replace Field with arc cotangent of Field.

Arc Sine  Replace Field with arc sine of Field.

Arc Tangent  Replace Field with arc tangent of Field.

Arithmetic Series  Fill Field with an arithmetic series from first record to last computed using the value n given in the Source / argument box for each step. If n = 5, Field will contain 0, 5, 10, 15, 20... series values in records from top to bottom.

Clear  Delete all values in Field.

Copy Day from  Examine the date/time in the Parameter field and copy the day number (as the day number in the month) to Field.

Copy Day of Week from  Examine the date/time in the Parameter field and copy the day number (as the day number in the week) to Field.

Copy Day of Year from  Examine the date/time in the Parameter field and copy the day number (as the day number in the year) to Field.

Copy from  Copy values from the given Parameter field to Field.

Copy Hour from  Examine the date/time in the Parameter field and copy the hour number (as the hour number in a 24 day) to Field.

Copy Leap Year Flag from  Examine a date/time field in the Parameter field, determines whether or not each date value falls on a leap year, and copies the result (0 or 1) into Field, which should be a numeric or Boolean field.

Copy Minute from  Examine the date/time in the Parameter field and copy the minute number (as the minute number in an hour) to Field.

Copy Month from  Examine the date/time in the Parameter field and copy the month number (as the month number in a year) to Field.

Copy Percentage from  Sum values in Parameter field and then for each record take the Parameter field's value as a percentage of the total and copy into Field.

Copy Record Deviation from  Find the median of all Parameter values and then for each record place in Field the deviation from that median for the particular Parameter value of that
Copy Record Variation from
Find the median of all Parameter values and then for each record place in Field the variation from that median for the particular Parameter value of that record.

Copy Running Product from
For each record in the sort order, form a product of Parameter field with the cumulative product of multiplying Parameter field from previous records. Place the result in the current record's Field.

Copy Running Sub from
For each record in the sort order, subtract Parameter field from the cumulative total of subtracting Parameter field from previous records. Place the result in the current record's Field.

Copy Running Total from
For each record in the sort order, add Parameter field to the cumulative total of subtracting Parameter field from previous records. Place the result in the current record's Field.

Copy Second from
Examine the date/time in the Parameter field and copy the second number (as the second number in a minute) to Field.

Copy Week from
Examine the date/time in the Parameter field and copy the week number (as the week number in a 52-week year) to Field.

Copy Year from
Examine the date/time in the Parameter field and copy the year number to Field.

Cosine
Replace Field with cosine of Field.

CoTangent
Replace Field with cotangent of Field.

Decimal Fraction
Replace Field with trailing decimal values of Field. 3.1415 would be replaced by 0.1415.

Divide by
Divides Field by the given Parameter value or field. Division by zero results in no change to Field for those records.

Exponentiate
Raise \( e \) to the power of the Field and place the result in the Field. (A true mathematician’s exponentiation)

Fill with
Fill the Fields of all records with what is in the Parameter. Equivalent to "Copy from" the Parameter value or field.

Fill with Average of
Examine Parameter field values and copy average of these to Field for all records.

Fill with E
Fill with \( e \), that is, 2.7182818285

Fill with Excess of
Examine Parameter field values and copy excess of these to Field for all records.

Fill with Maximum of
Examine Parameter field values and copy maximum of these to Field for all records.

Fill with Median of
Examine Parameter field values and copy median of these to Field for all records.

Fill with Minimum of
Examine Parameter field values and copy minimum of these to Field for all records.

Fill with Pi
Fill with \( \pi \), that is, 3.14159265359

Fill with Random
Fill Field with random numbers.

Fill with Skewness of
Examine Parameter field values and copy skewness of
these to Field for all records.

**Fill with StdDeviation of** Examine Parameter field values and copy standard deviation of these to Field for all records.

**Fill with Sum of** Examine Parameter field values and copy sum of these to Field for all records.

**Fill with Variance of** Examine Parameter field values and copy variance of these to Field for all records.

**Geometric Series** Fill Field with a geometric series from first record to last computed using the value n given in the Source / argument box for each step. If n = 2, Field will contain 1, 2, 4, 8, 16, 32 ...series values in records from top to bottom.

**Invert** Replace Field with (1 / Field). No change if Field contains zero.

**Logarithm (Binary)** Replace the value of the Field with the base 2 logarithm of that value.

**Logarithm (Decimal)** Replace the value of the Field with the base 10 logarithm of that value.

**Logarithm (Natural)** Replace the value of the Field with the natural logarithm of that value.

**Logarithm by** Replace the value of the Field with the logarithm of that value taken to base given in the Parameter field.

**Make Absolute** Take absolute value of the Field and place it in the Field.

**Make at Least** Resets all values in the Field which are lower than the field or value specified in the Parameter box to the field or value specified in the Parameter box. A lower bound clipping function controlled by the field or value specified in the Parameter box.

**Make at Most** Resets all values in the Field which are higher than the field or value specified in the Parameter box to the field or value specified in the Parameter box. An upper bound clipping function controlled by the field or value specified in the Parameter box.

**Modulo** Replace the Field with Field modulo value or field given in the Source / argument box.

**Multiply by** Multiplies Field by given field or value in Source / argument box.

**Negate** Multiply Field by -1 (switches sign).

**Power to** Raise Field value to the power specified by the given Parameter value or field.

**Round** Return the largest integer that is less than or equal to .5 + the value.

**Round Down** Returns the largest integer that is less than or equal to the value.

**Round Up** Returns the smallest integer that is greater than or equal to the value.

**Sine** Replace Field with sine of Field.

**Square** Replace Field with square of Field.

**Square Root** Replace Field with square root of Field.
### Subtract
Subtract the field or value specified in the Source / argument box from the Field.

### Tangent
Replace Field with tangent of Field.

### Date / Time Operators

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clear</td>
<td>Delete all values in Field.</td>
</tr>
<tr>
<td>Copy Day from</td>
<td>Examine the date/time in the Parameter field and copy the day number (as the day number in the month) to Field.</td>
</tr>
<tr>
<td>Copy Day of Week from</td>
<td>Examine the date/time in the Parameter field and copy the day number (as the day number in the week) to Field.</td>
</tr>
<tr>
<td>Copy Day of Year from</td>
<td>Examine the date/time in the Parameter field and copy the day number (as the day number in the year) to Field.</td>
</tr>
<tr>
<td>Copy Month from</td>
<td>Examine the date/time in the Parameter field and copy the month number (as the month number in a year) to Field.</td>
</tr>
<tr>
<td>Copy Second from</td>
<td>Examine the date/time in the Parameter field and copy the second number (as the second number in a minute) to Field.</td>
</tr>
<tr>
<td>Copy Week from</td>
<td>Examine the date/time in the Parameter field and copy the week number (as the week number in a 52-week year) to Field.</td>
</tr>
<tr>
<td>Copy Year from</td>
<td>Examine the date/time in the Parameter field and copy the year number to Field.</td>
</tr>
<tr>
<td>Decrease Day by</td>
<td>Decrement day by one (with rollover) in the date-time field.</td>
</tr>
<tr>
<td>Decrease Hour by</td>
<td>Decrement hour by one (with rollover) in the date-time field.</td>
</tr>
<tr>
<td>Decrease Minute by</td>
<td>Decrement minute by one (with rollover) in the date-time field.</td>
</tr>
<tr>
<td>Decrease Month by</td>
<td>Decrement month by one (with rollover) in the date-time field.</td>
</tr>
<tr>
<td>Decrease Second by</td>
<td>Decrement second by one (with rollover) in the date-time field.</td>
</tr>
<tr>
<td>Decrease Week by</td>
<td>Decrement week by one (with rollover) in the date-time field.</td>
</tr>
<tr>
<td>Decrease Year by</td>
<td>Decrement year by one in the date-time field.</td>
</tr>
<tr>
<td>Fill with</td>
<td>Fill the Fields of all records with what is in the Parameter. Equivalent to “Copy from” the Parameter value or field.</td>
</tr>
</tbody>
</table>
Fill with Maximum of Examine Parameter field values and copy maximum of these to Field for all records.

Fill with Median of Examine Parameter field values and copy median of these to Field for all records.

Fill with Minimum of Examine Parameter field values and copy minimum of these to Field for all records.

Increase Day by Increment day by one (with rollover) in the date-time field.

Increase Hour by Increment hour by one (with rollover) in the date-time field.

Increase Minute by Increment minute by one (with rollover) in the date-time field.

Increase Month by Increment month by one (with rollover) in the date-time field.

Increase Second by Increment second by one (with rollover) in the date-time field.

Increase Week by Increment week (with rollover) by one in the date-time field.

Increase Year by Increment year by one in the date-time field.

Make at Least Resets all values in the Field which are lower than the field or value specified in the Parameter box to the field or value specified in the Parameter box. A lower bound clipping function controlled by the field or value specified in the Parameter box.

Make at Most Resets all values in the Field which are higher than the field or value specified in the Parameter box to the field or value specified in the Parameter box. An upper bound clipping function controlled by the field or value specified in the Parameter box.

**Percentage Operators**

**Clear** Delete all values in Field.

**Copy from** Copy values from the given Parameter field to Field.

**Copy Percentage from** Sum values in Parameter field and then for each record take the Parameter field's value as a percentage of the total and copy into Field.

**Fill with** Fill the Fields of all records with what is in the Parameter. Equivalent to "Copy from" the Parameter value or field.

**Fill with Average of** Examine Parameter field values and copy average of these to Field for all records.

**Fill with Maximum of** Examine Parameter field values and copy maximum of these to Field for all records.

**Fill with Median of** Examine Parameter field values and copy median of these to Field for all records.

**Fill with Minimum of** Examine Parameter field values and copy minimum of these to Field for all records.

**Fill with Random** Fill Field with random numbers.
Logical And with Places a 0 or a 1 in the Field based on logical AND with the field or value specified in the Source / argument box. Any nonzero value counts as a 1 for the AND operation.

Logical Not If the Field contains 0, it is made a 1. If it contains any nonzero value it is made a 0.

Logical Or with Places a 0 or a 1 in the Field based on logical OR with the field or value specified in the Source / argument box. Any nonzero value counts as a 1 for the OR operation.

Logical Xor with Places a 0 or a 1 in the Field based on logical XOR with the field or value specified in the Source / argument box. Any nonzero value counts as a 1 for the XOR operation.

Make at Least Resets all values in the Field which are lower than the field or value specified in the Parameter box to the field or value specified in the Parameter box. A lower bound clipping function controlled by the field or value specified in the Parameter box.

Make at Most Resets all values in the Field which are higher than the field or value specified in the Parameter box to the field or value specified in the Parameter box. An upper bound clipping function controlled by the field or value specified in the Parameter box.

URL Operators

Append Appends given text or field to original field.

Clear Clear all values in Field.

Copy Day Name from Examine the date/time in the Parameter field and copy the name of the day to Field using local language settings.

Copy First Token from Copy the first token from Parameter into the Field. If Parameter contains John/Paul/George/Ringo the result would be to copy John into Field.

Copy Month Name from Examine the date/time in the Parameter field and copy the name of the month to Field using local language settings.

Copy URL Object from Copy main part of URL field.

Copy URL Parameters from Copy query parameters, if any, in URL.
Example: "?pg=q&sc=on" from http://www.mydomain.com/cgi-bin/query?pg=q&sc=on

Copy URL Port from Return port, if used, from URL string.
Copy URL Protocol from
Return protocol used by the URL.
Example: "http://" from
http://www.manifold.net

Copy URL Server from
Return the server from the URL. Eg:
"www.manifold.net" from

Decode with
Decrypt Field using a password given in Parameter.

Delete Containing
Clear values in original field that contain the given text or field.

Delete Containing Match
Clear all values that contain a match to the given regular expression pattern.

Delete Ending with
Clear values in original field that end with the given text or field.

Delete Ending with Match
Clear all values that end with a match to the given regular expression pattern.

Delete First Token
Delete the first token in the Field together with the separator (if any) following it. If applied to John/Paul/George/Ringo the result would be Paul/George/Ringo.

Delete Last Token
Delete the last token in the Field together with the separator (if any) preceding it. If applied to John/Paul/George/Ringo the result would be John/Paul/George.

Delete Left
Delete the first n characters of the original field, where n is a number specified by the given numeric field or value.

Delete Matching
Clear all values that match the given regular expression pattern.

Delete Right
Delete the last n characters of the original field, where n is a number specified by the given numeric field or value.

Delete Sounding Like
Clear all values that sound like the given word. Requires a lot of faith in the Soundex algorithm. A better idea is to use a viewbot to select all that sound like the given word and then take a look at the selection before deleting.

Delete Starting with
Clear values in original field that start with the given text or field.

Delete Starting with Match
Clear all values that start with a match to the given regular expression pattern.

Delete Starting with Token
Clear all values that start with the given token.

Encode with
Encrypt Field using a password given in Parameter. Caution: don't "Encode" a field twice by accident!

Fill with
Fill the Fields of all records with what is in the Parameter. Equivalent to "Copy from" the Parameter value or field.

Fill with Maximum of
Find the "maximum" value that occurs in Parameter field using lexicographic arithmetic and copy it to the Field.

Fill with Median of
Find the "median" value that occurs in Parameter field using lexicographic arithmetic and copy it to the Field.
<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fill with Minimum of</td>
<td>Find the “minimum” value that occurs in Parameter field using lexicographic arithmetic and copy it to the Field.</td>
</tr>
<tr>
<td>Leave Containing</td>
<td>Clear all values in original field that do not contain the given text or field.</td>
</tr>
<tr>
<td>Leave Containing Match</td>
<td>Clear all values except those that contain a match to the given regular expression pattern.</td>
</tr>
<tr>
<td>Leave Containing Token</td>
<td>Clear all values except those that contain the given token.</td>
</tr>
<tr>
<td>Leave Ending with</td>
<td>Clear all values in original field that do not end with the given text or field.</td>
</tr>
<tr>
<td>Leave Ending with Match</td>
<td>Clear all values except those that end with a match to the given regular expression pattern.</td>
</tr>
<tr>
<td>Leave Ending with Token</td>
<td>Clear all values except those that end with the given token.</td>
</tr>
<tr>
<td>Leave Left</td>
<td>Clip all text from the Field except the first n characters, where n is a number specified by the value given in the Source / argument box.</td>
</tr>
<tr>
<td>Leave Matching</td>
<td>Clear all values except those that match the given regular expression pattern.</td>
</tr>
<tr>
<td>Leave Right</td>
<td>Clip all text from the Field except the last n characters, where n is a number specified by the value given in the Source / argument box.</td>
</tr>
<tr>
<td>Leave Sounding Like</td>
<td>Clear all values except those that sound like the given word. Not for the faint of heart. A better idea is to use a viewbot to select all that sound like the given word and then take a look at the selection, invert the selection and then delete.</td>
</tr>
<tr>
<td>Leave Starting with</td>
<td>Clear all values in original field that do not start with the given text or field.</td>
</tr>
<tr>
<td>Leave Starting with Match</td>
<td>Clear all values except those that start with a match to the given regular expression pattern.</td>
</tr>
<tr>
<td>Leave Starting with Token</td>
<td>Clear all values except those that start with the given token.</td>
</tr>
<tr>
<td>Make Lower Case</td>
<td>Convert all characters in the Field to lower case.</td>
</tr>
<tr>
<td>Make Sentence Case</td>
<td>Capitalize first character in sentences. Example: “small green tree” becomes “Small green tree”</td>
</tr>
<tr>
<td>Make Title Case</td>
<td>Capitalize first character in words. Example: “small green tree” becomes “Small Green Tree”</td>
</tr>
<tr>
<td>Make Upper Case</td>
<td>Convert all characters in the Field to upper case.</td>
</tr>
<tr>
<td>Prepend</td>
<td>Prepends given text or field to original field.</td>
</tr>
<tr>
<td>Reverse</td>
<td>Reverses the order of characters in the Field.</td>
</tr>
<tr>
<td>Reverse by Tokens</td>
<td>Reverses the order of tokens in the Field. If applied to John/Paul/George/Ringo the result would be Ringo/George/Paul/John.</td>
</tr>
<tr>
<td>Reverse Token at</td>
<td>Take the token in the Field at the position given by the number in the value box and reverse its characters. If applied to John/Paul/George/Ringo with a value of 3 would result in John/Paul/egroeG/Ringo.</td>
</tr>
<tr>
<td>Span Excluding</td>
<td>Start at the first character in the Field and see if it occurs anywhere in the text string in the Source /</td>
</tr>
</tbody>
</table>
argument box. If it does not occur, continue to the next character and see if it occurs in the text string in the Source / argument box. Continue until a character is encountered that is in the text string in the Source / argument box and stop. Take the characters checked thus far in order and place them in the Field. Order in the Field is preserved, while the order of characters in the text string in the Source / argument box or field does not matter.

**Span Including** Start at the first character in the Field and see if it occurs anywhere in the text string in the Source / argument box or field. If it does occur, continue to the next character and see if it occurs in the text string in the Source / argument box. Continue until a character is encountered that is not in the text string in the Source / argument box and stop. Take the characters found thus far in order and place them in the Field. Order in the Field is preserved, while the order of characters in the text string in the Source / argument box or field does not matter.

**Toggle Case** Swap upper case for lower case throughout the text.

**Trim** Remove leading and trailing white space characters from the Field.

**Trim Left** Remove leading white space characters from the Field.

**Trim Right** Remove trailing white space characters from the Field.

**See Also**

*Transform - Using Tokens and Text Strings*
Transform - Using Tokens and Text Strings
This topic explains the use of Tokens with text operators.

A token is one or more characters that is found in a text field between token separator characters. Any character may be specified as a token separator character in the Tool Properties pane. By default, a variety of brackets and other punctuation marks are listed as token separators. For example, the semicolon (;) character is listed as a token separator by default. If you don't like the default list of token separators, you can specify your own separators by entering them into the Separate tokens with box in the Tool Properties pane.

Using the default token separator characters, the string John;Paul;George;Ringo would contain the tokens "John", "Paul", "George", and "Ringo". If this string occurred in a field in the value box, the text operator Copy First Token to would copy the string John into the subject field.

Suppose we have a table where dates are stored as text strings in the form 09/21/98 and we wish to extract separate month, day and year fields. Or, perhaps we have a table that stores text strings in the US format of month/day/year and we wish to convert to European format of day/month/year. Or, perhaps we wish to extract the year field so we can prepend a "20" to year dates such as "00" and "01". All of these cases can be managed using a slash (/) character as a token separator. In this case, the string 09/21/98 would consist of three tokens, "09", "21" and "98". (Note that if the data were stored as a Microsoft Access Date field, we could use Copy Day To and similar Commander operators to extract the date, year, etc.)

Token separators may be combined. Order is not significant. A token is parsed as one or more characters between any two (or preceding or following at the beginning and end of the string) characters listed as a token separator in the Options dialog. Extra token separators are ignored when they occur together.

So, if / and * are listed as token separators, /John*/Paul*/George*Ringo is still parsed as the Fab Four tokens. Constructions such as 30°42'13.0" can be parsed as three separate tokens if the *, ', and " characters are listed as token separators.

Caution: If we wish to parse as a single token numeric values that are written as strings using a decimal point, we must make sure to remove the period, ., from the list of token separators if we have added it to the list as a token separator. Otherwise, text strings such as "3.1415" will get parsed as two tokens, "3" and "1415"

Tip: how to extract the "nth" token? Repeatedly use the Move First Token To command n times to get to the nth token. For example, suppose we have a field called location that contains values in the form 30°42'13.0" and we wish to extract this information into three separate fields called degrees, minutes, and seconds. We would perform the following commands, where the Target box is in blue, the operator is in black and the source / argument box is in violet.

location Move first token to degrees
location Move first token to minutes
location Move first token to seconds

Tip: Don't be intimidated by the use of tokens. It's a lot easier than it appears from the above. Just dig in and try a few examples. Also, don't hesitate to change the characters listed in the Tool Properties pane as token separators. One often wishes to use only a single character as a token separator. This takes but a moment to change to suit any need.

See the Extract Last Names using Tokens example for a simple step-by-step procedure using tokens.
Transform - Span Excluding / Including

These functions in the Transform Toolbar for Tables are similar to the programming functions of the same name. They are simple to use even though they often seem too complex to learn. These examples show the Target box contents in blue, the operator in black and the Source / argument box contents in violet.

Span Excluding keeps going on until it finds something that is supposed to be excluded and then it stops. Span Including keeps going on as long as it keeps finding things that are supposed to be included. If it finds something not in the included list, it stops. A simple example that uses these functions to extract numbers from a text list of latitudes and longitudes will help us understand how these functions work.

Suppose we have columns of latitude and longitude coordinates written as text strings where each coordinate has a redundant "N" "S" "E" or "W" plus some commentary after the actual coordinate number. In the latitude field, a text field in this table, we encounter coordinates written as:

-45.8908S (verified 1982)

We’d like to strip out the trailing text and leave just the numbers:

latitude span excluding NSEW

results in "-45.8908" for the above value. The span excluding operator kept on going until it found something to be excluded (in this case, the S character) and then it stopped. So far, so good.

Now, suppose we realize we have some records of the form:

-45.8908 (verified 1982)

…where the N, S, E, or W were forgotten; therefore, the previous example does not always work. We would like to get any leading combination of numbers and stop as soon as we encounter any other character, even a space:

latitude span including +-.1234567890

…does the trick. In this case, the span including operator will keep on going so long as it keeps finding characters in the included list (plus and minus signs, a decimal point, and digits). As soon as it hits anything else it stops.
Queries

Queries

Queries are components that contain statements written in the database language SQL. Opening a query by double clicking on it will open the query in an editing window to show the SQL statement it contains.

Clicking on the query to highlight it in the project pane and pressing the Run button will execute the query and will display the results in the opened window. If there are any errors in the query text (such as typographical errors or SQL syntax errors) the query text will be opened in a query window and the first error located will be highlighted.

Queries are usually used to create tables that contain records and fields from other tables in the project. The SQL statement specifies what the table should contain and how the contents should be displayed. The simplest form of query chooses columns and records from one table and displays them.

Queries are easy to create. Use File - Create - Query to create a new query and then edit it by right clicking on it and choosing Open. Enter the SQL statement desired and close the query. To make it easier for us to write queries, Manifold will automatically color different parts of the query text in different colors.

Whenever the query is run it will execute the SQL statement and will display in the resultant table window whatever table results from the SQL statement. The resultant table may be used like any other.

SQL is an industry standard database query language used to fetch records from tables and to present those records with the fields desired. The result of an SQL query is what appears to be a new table. SQL may also be used to create new tables by combining existing tables, and may even be used to alter the structure of the database and the data it contains.

Note: For very simple, one line queries with drawings an alternative to using queries is to use the Query toolbar.

To create a new query:

1. Select File - Create Query command within main menu, or
2. Press the Create button in the project pane toolbar and select Query, or
3. Right click within the project pane or on any component and select Create - Query.
4. Open the query for editing as described below.
5. Write the SQL statement desired. Changes are made directly to the query. Make a copy of the query if you wish to be able to abandon the changes.
6. Close the query window when done. Run the query as described below.

To edit a query:

1. Select the query component in the project pane and press the Open button on the project pane toolbar, or
2. Right click the component and select Open, or
3. Double click the component.
4. Make any changes to the SQL text contained in the query.

To execute (run) a query:

1. Select the query component in the project pane and press the Run button on the project pane toolbar, or
2. Right click the component and select Run, or
3. Open the query for editing and select Query - Run from main menu.

To export a query as a table:
1. Select the query component in the project pane and press the Run button on the project pane toolbar, or
2. Right click the component and select Run, or
3. Open the query for editing and select Query - Run from main menu.
4. Choose File - Export - Table to export as a table.

Example

Let's begin by importing the Orders table from the sample Nwind.mdb database.

<table>
<thead>
<tr>
<th>Ship Name</th>
<th>Ship Address</th>
<th>Ship City</th>
<th>Ship Country</th>
<th>Freight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Around the Horn</td>
<td>120 Hanover Sq.</td>
<td>London</td>
<td>UK</td>
<td>4.45</td>
</tr>
<tr>
<td>Alfreds Futteriiste</td>
<td>Obere Str. 57</td>
<td>Berlin</td>
<td>Germany</td>
<td>18.59</td>
</tr>
<tr>
<td>Vaffeljernet</td>
<td>Smaglajet 45</td>
<td>Åhus</td>
<td>Denmark</td>
<td>20.12</td>
</tr>
<tr>
<td>Wartian HandKu</td>
<td>Tokkki</td>
<td>Culu</td>
<td>Finland</td>
<td>4.13</td>
</tr>
<tr>
<td>French S.p.A.</td>
<td>Via Monte Bianco 34 Torino</td>
<td>Italy</td>
<td>3.62</td>
<td></td>
</tr>
<tr>
<td>Furia Bacallau &amp; Fruto Jardins das rosas n.</td>
<td>Lisboa</td>
<td>Portugal</td>
<td>74.22</td>
<td></td>
</tr>
<tr>
<td>Seven Seas Imports</td>
<td>90 Wadhurst Rd.</td>
<td>London</td>
<td>UK</td>
<td>49.21</td>
</tr>
<tr>
<td>Wellington Importador</td>
<td>Rua do Mercado, 12 Resende</td>
<td>Brazil</td>
<td>31.54</td>
<td></td>
</tr>
<tr>
<td>LINQ-Delicatessas</td>
<td>Ave. 5 de Mayo Fori I. de Margari</td>
<td>Venezuela</td>
<td>102.59</td>
<td></td>
</tr>
<tr>
<td>Richter Supermark</td>
<td>Starenweg 5</td>
<td>Genève</td>
<td>Switzerland</td>
<td>50.87</td>
</tr>
<tr>
<td>PROSELLA-Restaurant</td>
<td>5ª Ave. Los Polos Gr</td>
<td>Caracas</td>
<td>Venezuela</td>
<td>17.67</td>
</tr>
<tr>
<td>Piccolo un mőrd</td>
<td>Gesweig 14</td>
<td>Salzburg</td>
<td>Austria</td>
<td>22.1</td>
</tr>
<tr>
<td>Filies hermanet</td>
<td>184 chemin de Titte</td>
<td>France</td>
<td>113.01</td>
<td></td>
</tr>
</tbody>
</table>

The Orders table has many fields related to orders in the sample database. We will create a table that shows the country names for all orders where the freight is greater than 50.

Using File - Create - Query we insert a new query into the project and then open it for editing by double clicking it in the project. We will write a simple SQL query:

```
Select [Ship Country] from Orders where [Freight] > 50;
```

SQL is not case sensitive (except for string literals and string operations), nor does SQL care if we write our statements in one line or in multiple lines or if we use indents. Extra parentheses are fine as well. We could write our queries with the terms all jammed together in one line or we can write nice, neat, understandable SQL statements like the above. It is always wise to establish a regular style and stick to it to enhance readability of SQL both for our colleagues as well as for future review by us.

Manifold query windows will automatically color different parts of an SQL query with differently colored text. This helps avoid typographical errors. If we would like to change the colors used by Manifold to color query syntax, the Tools - Options - Colors page includes entries for Query Comment, Query Keyword, Query Number, Query Operator, Query String and Query Text colors. The Tools - Options - Fonts page restricts the choices for the Query font to fixed size fonts.
The query appears in the project pane using the standard Microsoft icon for queries (these are also called "views" in Access). If we select the query in the Project pane and click the Run button in the toolbar, the query opens as a table:

The table created by the query works just like a regular table. For example, if we right click onto the Ship Country column head and choose Sort Ascending from the context menu we can sort the table by this column:

We can also use many other table window commands such as the Best Fit commands.
SQL is a straightforward, easy-to-use query language.

```
SELECT [Ship Name], [Ship Country], [Freight]
FROM Orders
WHERE [Freight] > 50;
```

If we would like to see additional fields in our table we can add them to the query as shown above.

The result of running the new query is a table with three fields instead of just one.

**Types of Queries**

There are two main types of queries: **select** queries and **action** queries:

- **Select** queries show results in tables and always begin with the SQL word **select**. Select queries choose data from one or more tables and display it in tabular form. Select queries do not change any data in the tables.

- **Action** queries do not display data but rather perform a task when they are run. Action queries change data in existing tables by appending, updating or deleting records. Action queries can also create new tables. Action queries start with words other than **select**. For example, an update query starts with the word **Update**.

See the Simple Queries topic for an introduction to simple select queries that display data from one table.

See the Queries Using Multiple Tables topic for instructions on creating queries that combine fields from multiple tables.

See the Action Queries topic for information on queries other than select queries.

**Dot Nomenclature**

"Dot" nomenclature allows us to refer to exactly a desired field within a given table even when other tables in our project might use the same field name in a different way. Dot nomenclature is simply the table name followed by a period "." and then the field name:

```
Customers.ContactName
```
The above refers to the **ContactName** field in the **Customers** table. If we are writing simple queries that involve only one table we need not use dot nomenclature. We could simply write **ContactName** as in the examples above. However, if we are working with more than one table we can always use dot nomenclature to clear up any ambiguities about which field is intended.

**Autocompletion**

Manifold will offer to autocomplete SQL functions and other parts of the query, popping up lists of key words and other useful items, to help guide us in writing the query. See the Autocompletion in Queries topic.

**Names in Brackets**

Some database management systems (including Manifold tables) allow field names and table names to contain the space character and other non-alphanumeric characters. Such field names should be enclosed in `[ ]` square brackets when written in queries. Simple field names that do not contain spaces or other unusual characters need not be enclosed in brackets.

To preserve maximum capability to export data to a wide variety of database systems we suggest the use of field names that do not contain spaces or other unusual characters.

Dot nomenclature and names in brackets can be combined. For example, one could write

```sql
Select * from [Mexico Table]
where [Mexico Table].BUSES_1991 > 3000;
```

**Case Sensitivity**

SQL itself is not case-sensitive, but string literals and string operations are. Hence, comparing a pair of string values which only differ by case for equality will return false. If you want to compare string values ignoring their case, use the `UCase` function.

**Selection and SQL**

Because many SQL queries begin with the word **Select** it is tempting to think of them as adding items to the selection. Although this is "selection" in the database sense of the word, the actual records and the objects with which they are associated may or may not be selected in the Manifold sense of the word depending on the setting of the **Automatically select query records** option in the Tools - Options - Miscellaneous dialog.

The default setting of this option is to automatically select records reported by a query. This makes it easy to use a query to select objects in a drawing. If automatic selection of query records is not desired, uncheck the option. If the option is not checked, records resulting from a query will simply be displayed in a table. We can then select them as desired. For example, we could choose **Edit - Select All** to select all the records in that table.

For reference information on SQL see the SQL Reference Guide information beginning with the SQL in Manifold System topic.

**VBScript Style Operators and Functions**

Manifold SQL supports use of numerous VBScript operators and functions. For example, `LikeX` is similar to the standard SQL `Like` operator but uses VBScript style syntax to match regular expressions.

The `RegExp` function returns values based on regular expressions. It transforms strings in a manner similar to the Manifold **Edit - Replace** dialog with the **Regular Expressions** checkbox turned on, so that:

```sql
RegExp("abc", "b", "d") returns "adc",
RegExp("abc", "[a-b]", "d") returns "ddc",
RegExp("abc", "[a-b]", "") returns "c",
RegExp("ab c", "([a-z]*) ([a-z]*)", "$1+$2") returns "ab+rc".
```
RegExp("ab c", "([a-z]*) ([a-z]*)", "$2 $1") returns "c ab",

For more on VBScript style operators and functions see the Expressions topic in the SQL Reference Guide.

**Manifold SQL and ANSI Syntax Option**

Manifold’s SQL is designed to be compatible with Microsoft SQL as implemented in the Microsoft "Jet" database engine. Manifold SQL is intended to be a superset of Jet SQL with features supporting ANSI SQL as well. SQL syntax as documented within the Microsoft Jet SQL Reference for Access 2000 is a practical approach to Manifold SQL.

Manifold SQL has also been extended with special functions to use the spatial and geocoding capabilities of Manifold. Note that the geocoding extensions to SQL will be available only if either the Manifold US street address geocoding database is installed or if Microsoft MapPoint has been installed. See the Spatial Extensions and Geocoding Extensions and Raster Extensions topics for a list of extensions to SQL in Manifold that allow creation of SQL queries that make spatial comparisons, work with rasters or employ the geocoding engine for work with street addresses.

In addition to being Microsoft Jet-like, Manifold’s SQL engine may be switched into an alternate mode where it parses SQL queries using pure ANSI syntax. To do so, open the query and in the View - Properties dialog check the **Use ANSI-compatible syntax** box. It is also possible to automatically set the **Use ANSI-compatible syntax** option for all new queries by checking the **Make new queries ANSI-compatible** option in the Tools - Options dialog.

When the ANSI option is turned off, line comments start after an apostrophe (‘) or double dashes (--) and always start after line comments. The names of tables and columns can be enclosed in square brackets ([ ]) or backward apostrophes (`). Strings are enclosed in quotes (") and times are enclosed in quotes (") or hashes (#).

When the ANSI option is turned on, line comments start after double dashes (--) and always start after line comments. The names of tables and columns can be enclosed in square brackets ([ ]) or quotes ("), or backward apostrophes (`), or apostrophes (‘) and times are enclosed in apostrophes (‘) or hashes (#).

Clearly, the safest course given that users might change the settings is to use double dashes (--) to mark line comments and to enclose the names of tables and columns in square brackets ([ ]) and to enclose times with hashes (#). The main difference between having the ANSI option off or on is that with it off strings are enclosed in quotes (") and with the ANSI option on strings are enclosed in apostrophes (‘).

**Lookup Values and CStr and CAST**

Invoking CStr on a lookup value or using CAST to convert the value to a string returns the descriptive name of the lookup value.

**Example**

We have a table T with a lookup column Region with values East and West. We want to select all records with the value of Region being East. We can do this with the following query:

```sql
SELECT * FROM [T] WHERE CStr([Region]) = "East";
```

**Example**

We have a drawing D and we want to select all areas it contains. We can do this with the following query:

```sql
SELECT * FROM [D] WHERE CAST([Type (I)] AS TEXT) = "Area";
```

**Use cache Option**

In the View - Properties dialog for a query the **Use cache** option allows us to control the caching behavior of queries.
The **Use cache** option applies to table queries, such as those using SELECT and TRANSFORM. When this option is turned on, the query may cache the resulting table between invocations provided its text and parameters have not changed. This is obviously good for performance when lots of data is involved in the query.

When the **Use cache** option is turned off, the query forcefully re-computes itself (re-computing any dependent queries and refreshing any linked tables) each time it is run from the project pane or from the query window. By default, the option is turned off.

The default setting of the **Use cache** option may be specified in the Tools - Options - Miscellaneous dialog via the **Make new queries cached** option.

**Queries and Linked Components**

In addition to using queries as dynamically refreshed tables, one can also import or link queries as drawings, images or surfaces to create linked components. See the Linked Drawings topic as well as the Queries and Images or Surfaces topic for details.

**Queries and Virtual Tables for Images and Surfaces**

Queries may use data from images or surfaces by using **virtual tables** for these components. See the Virtual Tables for Images and Surfaces topic for details.

**Tech Tip**

We may simultaneously open a query for editing with **Open** in one window and also run it in another window with the **Run** button. Any changes to the query's SQL statement made in the editing window will take effect in the opened viewing window when we next press **Run**. Note that for the **Run** button to work the query component must be highlighted in the project pane.

Suppose we create a query using data from **Mexico Table**, the table associated with our sample map of Mexican provinces. We can open the query in an editing window. We can also press the **Run** button to execute the query and open the results in a table window. The screen shot above shows the editing window, the table window and the project pane. (Windows have been resized to very small size to fit into the screen shot).
We can change the 3000 value to 4000 in the SQL statement and then press the Run button. The table window will be updated with the new results of the query.

If we change the 4000 to 5000 we can press the Run button again to update the table once more. Note that there is no need to "save" the SQL statement in the editing window. Changes to the query in the editing window take effect whenever the Run button is pressed.

As a convenience for users, if a query is open at the same time as an editable query window and also as a table window reporting the results of the query being run, the Windows dialog will append the postfix (Table) to the latter. For example, if we have a query called MyQuery and it is open both as an editable query window and also as a table window reporting the results of running the query, the Windows dialog will show two windows open, one called MyQuery and the other called MyQuery (Table).

**Keyboard Shortcuts for Editing Queries**

In addition to the usual Windows keyboard shortcuts query windows support key combinations for doing advanced syntax-aware editing tasks:

- **CTRL-J** Display a list of auto-complete suggestions for the context word.
- **CTRL-K** Comment selected text.
- **CTRL-SHIFT-K** Uncomment selected text.
Queries

CTRL-M Convert selected text to lower case.
CTRL-SHIFT-M Convert selected text to upper case.
CTRL-Y Multi-level Redo.
CTRL-Z Multi-level Undo.
CTRL-] Check the character near the cursor and, if it is a bracket character, jump to the matching bracket.
CTRL-SHIFT-] Check the character near the cursor and, if it is a bracket character select the text between the bracket and the matching bracket.
CTRL-Space Auto-complete the context word.

Recognized bracket combinations are { and }, [ and ], { and }, ' and ', " and " and # and # (used to delimit dates in queries).

Query windows are unusual in that they support multi-level Undo / Redo. One can CTRL-Z (Undo) backwards through many changes and CTRL-Y (Redo) forwards to redo many Undo operations.

Formatting Columns

Queries also use the Formatting Columns capabilities of table windows, inheriting the format style used for the column in the table. Styles should be chosen to provide clarity of results.

Users are sometimes surprised to see numbers they expect to be values like 1.1 appear in a query or other table as 1.0999942779541, even if a floating point type is used. That is a result of the fundamental architecture of computers combined with the choice of format style used.

Floating-point values have finite precision. Some values, such as 1.2, or 0.25 can be represented precisely, but most can not. For example, neither 12.6 nor 13.7 can be represented precisely. To see what really hides behind a value shown as 12.6, set the format style of the relevant column to 12.34 and increase the number of decimal digits to 10. Try the same for 13.7.

If we subtract one approximation from another, as in the case of an SQL query, we get an approximation of the result so that an expected result such as 1.1 may appear as 1.0999942779541.

The default format style uses a Microsoft Windows API which makes some approximations, such as 12.6000..., look round but leaves others unaffected. There is some logic behind that API, but that logic is obviously not applicable to all cases. To provide fine control over desired appearance is one reason why Manifold allows setting the format style and the number of decimal digits explicitly.

Notes

See the Query Templates and Sample Queries example topics for samples of queries.

For very simple, one line queries with drawings an alternative to using queries is to use the Query toolbar.

The idea of relational databases originated in a 1969 IBM research report written by Dr. E.F. "Ted" Codd, who invented the relational model of databases and introduced it to the world. SQL originated as SEQUEL (for "Structured English Query Language") in 1974 at the IBM San Jose Research Laboratory in a project led by Donald Chamberlin.

Just about everyone in computing thinks that "SQL" is an acronym for "Structured Query Language." That’s not true, despite the historical origin of the language, as the ANSI standard defining SQL officially names it “Database Language SQL.” Experts will also point out that SQL is not structured, it is used for more purposes than just asking about things (the conventional meaning of the English word “query”) and it is technically not a “language” in the Turing sense of the word.
According to ANSI, “SQL” should be pronounced by saying the letters as in “ess cue ell.” However, many people prefer to pronounce the term as the English word “sequel,” especially when referring to Microsoft’s SQL Server DBMS product since “sequel server” is more alliterative and easier to say than “ess cue ell server.” It is very common to use both pronunciations, even in the same sentence, as in “This ess-cue-ell works in Sequel Server but not in Oracle.”

When running SQL within Manifold queries, one is using the Manifold SQL engine. When executing SQL within the Database Console one is using whatever SQL is the native SQL of the external database system. One should be aware that SQL implementations in various databases systems can contain bugs. If an SQL bug occurs within the Database Console, the bug should be tracked down with the vendor of the external database system being used.

Queries made with query components as well as queries made with the Query toolbar execute within Manifold on the local machine and thus must fetch the content of involved tables from their respective data sources. This frees the user from the hassles of different dialects of SQL as supported by different drivers, allows querying data sources that do not support any SQL at all and allows using spatial, geocoding and other extensions implemented within Manifold.

To make a query execute on a database server using that server’s own SQL, create a view on that server and link that view to the Manifold project as you would link a table.

Queries made within the Database Console run within the database server.

See Also

Autocompletion in Queries
SQL in Manifold System
SQL Reserved Words / Index
Expressions
View - Refresh Data
Spatial Extensions
Geocoding Extensions
Using SQL to Select Map Objects
Sample Queries
Query Templates
Autocompletion in Queries

When editing queries, Manifold will offer to autocomplete SQL statements, clauses and operators as well as table names and columns, popping up lists of key words and other useful items, depending on the context, to help guide us in writing the query. Autocompletion provides several key benefits:

- Increased editing speed.
- Fewer typographical errors.
- Consistent capitalization style
- Quick reference to component, table and column names in the project.
- Quick reference to SQL keywords, statements, clauses, functions and operators.

The default behavior is to pop up a list of possible completions when a candidate word is started or when a left bracket, [ character, is entered to signify the possible desire to use a table or column name. A list of completion candidates will pop up if the Automatically list completion suggestions when editing queries box in Tools - Options - Miscellaneous has been checked.

Pressing the keyboard down arrow moves the focus to the list. Pressing Enter, tab, or typing a bracket character inserts the selected suggestion into the query.

In addition, pressing CTRL-space after beginning a candidate word will have Manifold automatically complete the word if an unambiguous match is found in SQL key words.

These functions may be manually invoked by use the SQL autocompletion commands in the Edit - Advanced menu of commands available when a query is edited:

- **Complete Word**
  Appears for queries. Complete the word automatically if an unambiguous choice is possible. If no unambiguous choice is possible, pop up a list of choices (the same as Complete Word with List). Usually invoked with the standard Microsoft CTRL-space shortcut.

- **Complete Word with List**
  Appears for queries. Pop up a list of possible completions. Not usually used, since checking (the default) the Automatically list completion suggestions when editing queries box in Tools - Options - Miscellaneous causes the list to pop up automatically.

We can invoke the above manually with a CTRL-space and CTRL-J keyboard shortcuts.

To choose an autocompletion item presented to us we can double-click with the mouse or press the tab key if a choice is unambiguous. Manifold provides different pathways to autocompletion so that we can select whichever method is the fastest and most convenient at the time.

**Example**

Let's write a simple query using the Mexico sample drawing and table. This example assumes the Automatically list completion suggestions when editing queries box in Tools - Options - Miscellaneous is checked (the default setting). The query we will write is a single line of SQL:

```
SELECT [Place_name] FROM [Mexico Table] WHERE [POBL_1990] > 1000000;
```

Autocompletion will help us write the above query without the tedium of having to manually write out all the column and table names manually. Autocompletion also spares us much trouble caused by typographic errors that often creep into manually-entered text.

We begin by creating a new query and then double-clicking the new query open in a query window.
We enter the character $S$ (beginning the first character in the \texttt{Select} statement) and immediately the system pops open a menu of choices for autocompletion, scrolled down to the items that begin with $S$.

One way to autocomplete the \texttt{Select} statement is to double click the \texttt{SELECT} entry with the mouse.
This will immediately add the remaining characters in the `SELECT` statement to the query. Let us continue the query to specify what field we want to select.

We do this by entering the `[` left bracket character. The system immediately pops open a list of table names and column names available in the project. We double click the `Place_name` table to have it autocompleted.

The system automatically adds characters to complete the name of the column. About now the novice SQL programmer begins to realize that autocompletion is one of those wonderful things in life that once tasted cannot be done without.

There is a potentially faster way of invoking autocompletion than double-clicking with the mouse, as will be seen as we continue the query by building a `FROM` clause.
As we enter the F character to begin the FROM clause, the system pops up a menu of options that begin with F. Note that none of them is yet highlighted.

As we continue and enter the R character the system highlights FROM since that is the only match that begins with FR. At this point we can hit the tab character (a typical Windows shortcut for accepting an autocompletion choice)....
...and the system will complete the FROM statement for us.

It is true that entering a sequence of an F character and an R character followed by the tab character is not much of a time saver for an accomplished touch-typist who could just as rapidly bang out FROM using all four characters; however, in the case of more complex constructions, such as long field names, using autocompletion can be a great time saver as well as a way of guarding against typographic errors.

We can see the above effect by continuing the example.

If we begin entering a column name by entering M the system will jump to those columns that begin with M. We could double-click onto a column to choose it for autocompletion, or we could continue entering characters to get a highlighted match to an unambiguous autocompletion choice.
When we enter `[Mexico T` we get a highlighted match to the `Mexico Table` column, at which point we can press the `tab` key to autocomplete it.

Again, it's true that an accomplished touch-typist could enter the above text almost as rapidly as autocompleting it; however, by using autocompletion we guarantee the field name does not contain a typographic error because if it did a highlighted match for autocompletion would not have occurred. Saving just a single typographic error in a query can save us as much time in debugging as entering hundreds of characters.

Another advantage to autocompletion is consistent use of case. Because autocompletion uses the same capitalization style regardless of whether initial characters were uppercase or lowercase, autocompletion is also a handy way of adhering to a consistent capitalization scheme without having to pay too much attention to using the `SHIFT` key.
Queries

For example, suppose we continue the query by entering `WHERE` and then press the `tab` key for an autocompletion.

```
SELECT [Place_name] FROM [Mexico Table] WHERE
```

The system will enter a `WHERE` clause using all uppercase letters.

```
SELECT [Place_name] FROM [Mexico Table] WHERE [POB1_1990] > 100000;
```

We can finish the query by manually entering the remaining text.

**Options**

As mentioned in the beginning of the example, the above example assumes the **Automatically list completion suggestions when editing queries** box in Tools - Options - Miscellaneous is checked (the default setting). This option causes a list of choices to automatically appear whenever a SQL key word is begun or a `[` left bracket indicates the beginning of a table name or column name.

It is not to everyone’s taste to have a list of autocompletion choices appear automatically. If we don’t like the pop up lists, we can disable them by unchecking the option in Tools - Options. With the option turned off at any time we can cause a list to appear (if one is possible) by pressing `CTRL-J`.

For example, if the option is turned off and we enter...

```
SELECT [Place_name]
```
...and we would like to be reminded of what might come next we can always press CTRL-J and a list of SQL statements, clauses and operators will appear.

Note that regardless of whether the above option is turned on or off at any time when beginning an SQL key word we can have it completed for us with a CTRL-space if we have entered enough characters to unambiguously select a possible completion choice.

See Also

Edit - Advanced
Queries
Simple Queries
**Simple Queries**

This topic provides an introduction to writing simple queries using SQL. For very simple, one line queries with drawings an alternative to using SQL queries is to use the Query toolbar.

Simple queries will display data from a single table. They use elementary SQL using the SELECT statement written as follows:

```
SELECT <fields> FROM <table> WHERE <condition>
```

A simple example using the **Customers** table from the **Nwind.mdb** sample database would be:

```
SELECT [Contact Name], [Phone] FROM [Customers]
WHERE [Country] = "France"
```

If we’ve imported the **Customers** table into our project and place the above SQL statement into a query, when we open the query we will see a simple table with two columns:

<table>
<thead>
<tr>
<th>Contact Name</th>
<th>Phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Daniel Torini</td>
<td>30.59.64.10</td>
</tr>
<tr>
<td>Frédérique Citeaux</td>
<td>68.60.15.31</td>
</tr>
<tr>
<td>Martine Rancé</td>
<td>23.16.10.16</td>
</tr>
<tr>
<td>Janine Labrune</td>
<td>40.67.86.98</td>
</tr>
<tr>
<td>Arnette Roulet</td>
<td>61.77.61.10</td>
</tr>
<tr>
<td>Laurence Leibhan</td>
<td>91.24.45.40</td>
</tr>
<tr>
<td>Paal Hvorrsh</td>
<td>26.47.16.10</td>
</tr>
</tbody>
</table>

SQL is not case sensitive except inside text strings used within quotes in criteria. We can use parentheses to group criteria within criteria to keep our statements legible, and we use tabs and spaces, multiple lines and indents as desired to keep things clear.

**Templates**

```
SELECT <fields>
FROM <tables>
WHERE <condition>
GROUP BY <fields>
HAVING <criteria>
ORDER BY <expression>;
```

The Query Templates example topic provides a list of templates like the above. These are intended to guide creation of frequently occurring types of queries by providing a template for the SQL statement to be used. Replace the generic items in angle < > brackets with the specific items you wish to use.

The above template shows the basic options in simple queries. Choices except the **Select** and **From** constructs are optional. Click on the links above to jump to the applicable Help topic for more information.

**Autocompletion**

Manifold will offer to autocomplete SQL functions and other parts of the query, popping up lists of key words and other useful items, to help guide us in writing the query. See the Autocompletion in Queries topic.

**Tech Tip: SQL and Strings**
SQL is case insensitive for SQL itself but is case sensitive when evaluating string values. Suppose, for example, we have a query such as

\[
\text{SELECT } f \text{ FROM table WHERE } f = "13AF";
\]

This query will return no results if the value in \( f \) is \texttt{13aF}, because "13aF" is not the same as "13AF".

To normalize case, use the Transform toolbar Make Lower Case or Make Upper Case operators to force all values in the column to upper case or lower case. Alternatively, use a modified SELECT statement:

\[
\text{SELECT } f \text{ FROM table WHERE UCase(f)="13AF"};
\]

When comparing strings, SQL is sensitive to leading or trailing whitespace. "13AF  " does not equal "13AF". To remove leading / trailing whitespace use the Transform toolbar Trim Left, Trim Right or Trim operators to eliminate leading, trailing or both leading and trailing whitespace. Alternatively, use a modified SELECT statement:

\[
\text{SELECT } f \text{ FROM t WHERE Trim(f)="13AF"};
\]

How can we create strings with special characters in them? Use the \texttt{CHR} function. For example, to create a string that includes a quote character we can use

\[
"ab" + \text{Chr}(34) + "cd"
\]

which results in a string with an \texttt{ab"cd} value.

\textbf{See Also}

- Autocompletion in Queries
- SQL in Manifold System
- SELECT Statement
Queries Using Multiple Tables

Queries can incorporate fields from two different tables. There are several methods for doing so depending on the structure of the tables and their contents.

This is the easiest situation is when the two tables have a key field in common. We can then combine them by using an inner join between the tables. This is also called an equi-join in some database management systems.

```
SELECT [Customers].[Company Name], [Orders].[Employee ID]
FROM [Customers] INNER JOIN [Orders]
ON [Customers].[Customer ID] = [Orders].[Customer ID]
```

The inner join operation combines two tables (in the above case, the Customers table and the Orders table) using a key field they have in common. The example shown above will create a table that takes the Company Name field from the Customers table and shows it with the Employee ID field from the Orders table. It will create a table for all records where there are both customers and orders.

We can also create a similar query that will show all customer records in the results even if there are no orders for some customers. We might also want to run the query in a way that includes all records in the orders table even if there are no customers for some orders. This should not occur (a well-run business should not be recording orders for which there are no customers), but it is a way of checking for errors. To do this, we use an "outer join" which is phrased as either a left join or a right join.

```
SELECT [Customers].[Company Name], [Orders].[Employee ID]
FROM [Customers] LEFT JOIN [Orders]
ON [Customers].[Customer ID] = [Orders].[Customer ID]
```

The left join operation will include all of the records from the left-hand table mentioned, the Customers table in the example above, even if there are no matching values to records in the right hand table (the Orders table in the example above).

```
SELECT [Customers].[Company Name], [Orders].[Employee ID]
FROM [Customers] RIGHT JOIN [Orders]
ON [Customers].[Customer ID] = [Orders].[Customer ID]
```

The right join operation will include all of the records from the right-hand table mentioned, the Orders table in the example above, even if there are no matching values to records in the left hand table (the Customers table in the example above).

**Simplified Syntax for Inner Joins**

Inner joins can be created by default in Manifold using a simplified syntax such as the following:

```
SELECT [Customers].[Company Name], [Orders].[Employee ID]
FROM [Customers], [Orders]
WHERE [Customers].[Customer ID] = [Orders].[Customer ID]
```

This simplified syntax replaces the inner join and on words with a comma in the list of tables involved and a simple where clause. Many SQL systems can use the above syntax. Access uses the inner join syntax because the "inner join" construct is used as a hint to the Jet database engine within Access to build temporary indices that will be used to optimize the query. Manifold SQL optimizes both inner join and where constructs so the above simplified syntax works just as fast as using inner join.

SQL can also be used to combine fields from more than two tables, and even from multiple tables that do not have a key field in common. However, such SQL topics are beyond the introductory level of this help documentation. For advanced SQL discussion we suggest readings in one of the many good books oriented to SQL in a Microsoft Windows environment.

**The Identity Command**
Right clicking on a column head in a table created by a query pops open a context menu from which we can choose the **Identity** command. The **Identity** command allows us to select the ID column to be used in tables that have more than one identity column.

Suppose we have two drawings, one of cities and one of states, called **Cities** and **States**. The following query...

```sql
SELECT * FROM Cities, States
WHERE Contains(States.ID, Cities.ID);
```

...will generate a table with two identity columns: **States.ID** and **Cities.ID**.

Suppose the query generates a record containing a **New York** city and a state of **New York**. Let's also suppose that **New York** city is selected and the state of **New York** is not. The selection state of the record that is linked to both of these objects is determined by the active identity column. The record will appear selected if the active identity column is set to **Cities.ID** and unselected if the active identity column is set to **States.ID**.

The active identity column also specifies how selection in the table selects objects in the associated drawings. If the **Cities.ID** column is made the active identity column then selecting a record in the table will select the corresponding object in the **Cities** drawing. If the **States.ID** column is made the active identity column, then selecting a record in the table will select the corresponding object in the **States** drawing.

**Note**

When removing components contributing to a content of a query Manifold tries to preserve the query output if a query table is open. Suppose we select data from two tables in a query and the resulting query table shows some columns from the first table and some columns from the second table. If we delete the first table while the query is open, the cells of the resulting table that belong to the first table will become empty but the cells that belong to the second table will stay populated. Removing all components contributing to the content of a query will close the query table.

**See Also**

- SQL in Manifold System
- SELECT Statement
- CROSS JOIN Operator
- OUTER (LEFT, RIGHT, FULL) JOIN Operators
Calculations in Queries

Queries can automatically perform calculations to create new columns from existing fields. The general syntax is

```
SELECT <expression> [AS <name>] FROM <table>;
```

For example,

```
SELECT [Unit Price] * [Quantity] * (1 - [Discount]*100)/100
AS [Total Price] FROM [Order Details]
```

Would create a table with one column called Total Price. The value for each record in Price would come from multiplying the Unit Price, Quantity and normalized Discount fields in the Orders table. (Note that using square brackets for field names is optional except for Unit Price, which has a space in the field name.)

We can use multiple field expressions in the Select construct's field list by separating each with a comma:

```
SELECT [Unit Price] * (1 - [Discount]*100)/100
AS [Discounted Price],
[Unit Price] * [Quantity] * (1 - [Discount]*100)/100
AS [Total Price] FROM [Order Details]
```

This creates two columns called Discounted Price and Total Price. Note that queries are insensitive to extra white space or line breaks so that we can add extra line breaks in the above to make it fit within the narrow margins of this document and still be reasonably legible.

Operators

Manifold queries can utilize numerous operators and expressions. These include arithmetic, comparison and logical operators. See the Expressions topic.

Examples

```
SELECT ([First Name] & " " & [Last Name]) AS [Full Name]
Creates a column called Full Name that contains the contents of the First Name field plus a space plus the Last Name field contents.

SELECT (City & ", " & State & " " & Zip) AS Address
Creates a column called Address that contains the contents of the City field plus a comma and a space plus the State field, a space and the Zip field contents.

SELECT (Mid(Phone),2,3) AS [Area Code]
Create a column called [Area Code] that contains the second, third and fourth characters of the Phone field. If phone numbers are in the US form of (800)556-5919 this yields the area code. The Mid function's parameters ask it to report the three characters beginning with the second character. [Only works if we are sure all phone numbers are in the above form!]

SELECT [Freight] * 1.1 AS [Freight Charged]
Create a Freight Charged column that contains 1.1 times the value of the Freight column.

SELECT [Quantity] * [Unit Price] AS [Amount]
Create an Amount column that multiplies Quantity by Unit Price for each record.

SELECT DateDiff("d", [Required Date], [Shipped Date]) AS [Lead Time]
Creates a Lead Time column by subtracting the Shipped Date value from the Required Date value. Note that use of parentheses is optional but that it helps legibility.

1375
SELECT [Units in Stock] + [Units on Order] AS [Total Stock]

Show a Total Stock column that adds the Units in Stock and Units on Order fields.

SELECT DateDiff("d", [Order Date], [Shipped Date]) AS [Lag Time]

Show a Lag Time column that uses the DateDiff function to compute the number of days between the order and shipping dates.

SELECT DatePart("yyyy", [Hire Date]) AS [Year Hired]

Create a column showing the year hired by using the DatePart function to find the year from the Hire Date field.

See Also

SQL in Manifold System
SELECT Statement
Expressions
Action Queries

Action queries are queries that make changes to many records at once. They are used to delete records, to update records (that is, to change values in them), to create new tables, to delete tables and to launch queries that accept a user-supplied parameter. Action queries are just like any other SQL query in that they use SQL and are written within a Manifold query component. They are given the special name "action" to highlight their activity in changing the tables or interacting with the user by virtue of the SQL commands they use.

Examples

Using the Nwind.mdb sample database:

```
DELETE FROM [Order Details] WHERE [Discount] = 0;
```
Delete records for non-discounted orders.

```
DROP TABLE [Categories];
```
Delete the Categories table (square brackets are not necessary).

```
PARAMETERS Threshold INTEGER;
SELECT * FROM [Order Details] WHERE Quantity > Threshold;
```
Select orders whose quantity exceeds a number entered by the user.

```
SELECT DISTINCT [City] INTO [Locations] FROM [Customers];
```
Select all cities from Customers into the new table (Locations). The use of the DISTINCT quantifier suppresses duplicates.

```
UPDATE [Order Details] SET [Unit Price] = [Unit Price]*0.95 WHERE Quantity >= 30;
```
Lower prices (by 5%) on items whose quantity is too high.

Notes

Running an action query reports the number of affected records in the History pane.

See Also

Parameter Queries for an example of writing and launching a parameter query.

SQL in Manifold System
ALTER TABLE Statement
CREATE TABLE Statement
CREATE VIEW Statement
DELETE Statement
DROP TABLE Statement
DROP VIEW Statement
INSERT INTO Statement
SELECT INTO Statement
UPDATE Statement
Parameter Queries

Queries using PARAMETERS can ask for user specified parameters when the query is run. Parameter variables are declared in the first line of the query using a declaration in the form:

```
PARAMETERS name type;
```

Where `name` is a legal variable name (no spaces or special characters) and `type` is a valid data type such as TEXT, INTEGER, etc. Thereafter, the parameter variable can be used in the query as a variable.

The project pane will report the number of parameters in a query when the query is highlighted in the project pane.

Example

This example uses the Order Details table from the Nwind.mdb sample database. Import this table from the database.

Use File - Create - Query to create a query and then enter the above text.

Press the Run button in the project pane to execute the query.

When we do so, the Query Parameters dialog launches to allow us to enter values for those parameters used in the query. The query we've written is a simple query that has only one user supplied parameter called Threshold so only this parameter appears in the dialog.

The first row in the list of query parameters (if there is more than one) will be highlighted. To enter a value, double click into the Value cell for that parameter, enter the desired value and then press Enter.
We've entered the value 10. Press OK to continue with the query.

Queries using more than one parameter can be constructed. List multiple parameters separated with commas, for example:

PARAMETERS [Beginning Date] DateTime, [Ending Date] DateTime;

When more than one parameter has been used the Query Parameters dialog will show each parameter. To specify the values desired for each parameter, double click into the Value cell for each parameter and enter the value desired.

Tech tip: a keyboard shortcut to double clicking into a Value cell is to use the down and up keyboard arrows to highlight the desired row in the dialog and then press the F2 key to begin editing the value.

See Also

Queries
SQL in Manifold System
Data Types
PARAMETERS Declaration
Selecting Objects with Queries

This topic explains how to select objects in drawings (or in drawing layers in maps) using queries. Queries can be used to select objects either automatically or manually using mouse-based selection commands within the query table.

Queries are written using SQL, which usually begins with the SELECT command. This meaning of "selection" is the traditional database meaning of picking out records and columns from database tables to construct a new table. It is a different meaning than the Manifold use of the word "selection" to mean choosing objects in drawings or pixels in images or otherwise choosing items in other components.

The normal use of queries is to generate a table display showing the results of the query. If the query includes the ID field in its results, query tables can automatically select the drawing objects corresponding to each row in the query table based upon the setting of the Automatically select query records option in the Tools - Options - Miscellaneous dialog. When this option is checked (the default setting) queries will automatically select objects in the drawing.

That is, with this option on querying a drawing will automatically deselect any selected objects and then select all objects returned by the query. This is equivalent to making a selection using Replace mode. When a query includes more than one drawing, all previously selected objects in all of the drawings will first be deselected and then objects from the query will be selected in the first drawing for which an ID column is found in the query.

If the Automatically select query records option is not checked then running a query will generate a tabular display of the query results, but the objects in that query will not automatically be selected. We can then use such a query-generated table to manually select objects in the drawing with interactive mouse commands much as we could use a drawing's table to select objects in the drawing.

Use the manual option if queries will be used to make incremental, interactive selections, for example using Add or Subtract selection modes with mouse selection within the query window to add the query's results to a previous selection or to subtract the query's results from a previous selection.

Query Selection in Drawings

There are three cases of queries to consider:

- Queries using one drawing that result in a single ID column in the query table.
- Queries using two or more drawings that result in multiple ID columns in the query table.
- Queries joining columns from different tables.

Queries using One Drawing

If the Automatically select query records option is checked, queries that reference one drawing may be used to select objects in that drawing if they include the ID field in the query. If the automatic option is not checked, the query table can be used for interactive selection using mouse commands (clicking on records) just as selection operates with the drawing's table.

```sql
SELECT * FROM [Mexico_eg Drawing]
WHERE [Place_name] LIKE "BAJA%";
```

The above query, for example, will create a table from the example mexico_eg.mif drawing that finds all Mexican provinces with "BAJA" in their name. If the Automatically select query records option is checked the query will also select all objects that have "BAJA" in their name.

If the Automatically select query records option is not checked the query table will display but no objects will be selected. If desired we can select all of the objects in the query table with mouse selection commands. For example, we could click on the first record in the query table and SHIFT click on the last record in the query table to select all of the items in the query.

We can use either a drawing or the drawing's table in a query to exactly the same effect. For example, the above query could have been written...
```
SELECT * FROM [Mexico_eg Table]
WHERE [Place_name] LIKE "BAJA%";
```

...with exactly the same effect.

Objects are not selected by queries in Manifold unless the ID system field is selected by the query. The ID field is required either for automatic selection or for manual selection using mouse commands. For example, if we have a drawing of provinces that each has a name, the SQL fragment ...

```
SELECT Name from [Provinces Table] WHERE...
```

... will create a table, but it will not select the objects associated with the records that appear in that table. In contrast, the fragment...

```
SELECT ID, Name from [Provinces Table] WHERE...
```

... will select the objects as well as creating the table.

**Queries using Two Drawings**

Queries that reference two drawings may be used to select objects in either drawing using the Identity command.

Suppose, for example, that we have created a map with a drawing of points, called Points, as a layer above the Mexican boundaries map, which we have renamed Mexico for brevity.

```
SELECT * from Mexico, Points
WHERE Contains(Mexico.ID, Points.ID);
```

The above query selects records in both Mexico and Points for those cases where provinces in Mexico contain some of the points. To change which drawing's records are displayed in the query, choose the Identity command in the column context.

If the Automatically select query records option is checked, the query will select objects from the drawing (in this case Mexico) for which the first ID column appears.

Suppose we have two drawings, one called Points that contains points and one called Areas that contains areas. We select some areas in the Areas drawing and now we would like to use a query to select all points in the Points drawing that are inside the selected areas:

```
SELECT * FROM [Points], [Areas]
```
WHERE [Areas].[Selection (I)]
AND Contains([Areas].[ID],[Points].[ID]);

Note the use of the Selection (I) intrinsic field to use only selected areas in the query.

Queries Joining Columns from Different Tables

In current editions of Manifold, queries that join columns from tables not associated with drawings cannot be used to select objects in drawings. For example, a query formed with "Union D1, D2, T3" where D1 and D2 are drawings and T3 is a table could not be used to select objects in the drawings.

Selection Modes and Selection from Queries

Selection modes do not affect automatic selection when the Automatically select query records option is checked. Automatic selection always operates in Replace mode.

The selection mode set affects selection only within the query window using interactive mouse commands.

For example, if we have Select Replace set (the default, seen above) any selections we make in the query window will replace the entire selection in the query window. This will change the selection in the referenced drawing window only to the extent that the objects in the drawing are represented in the query window.

Suppose that in the example seen above we have previously selected the province of Oaxaca in the southern part of Mexico.

We can then run the following query to find all provinces in Mexico that contain one of the points in the Points layer.

```
SELECT DISTINCT Mexico.ID, Mexico.Place_name from Mexico, Points
WHERE Contains(Mexico.ID, Points.ID)
```

We can run this query by pressing the Run button in the project pane toolbar.
The result will be a table listing the IDs and names of each province (seen with Identity set to the Mexico drawing).

<table>
<thead>
<tr>
<th>ID</th>
<th>Place_name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SAN LUIS POTOSI</td>
</tr>
<tr>
<td>2</td>
<td>ZACATECAS</td>
</tr>
<tr>
<td>3</td>
<td>DURANGO</td>
</tr>
<tr>
<td>5</td>
<td>SINALOA</td>
</tr>
<tr>
<td>7</td>
<td>NUEVO LEON</td>
</tr>
<tr>
<td>11</td>
<td>COAHUILA</td>
</tr>
<tr>
<td>12</td>
<td>CHIHUAHUA</td>
</tr>
<tr>
<td>30</td>
<td>NAYARIT</td>
</tr>
</tbody>
</table>

With the focus on the query table window we can select all the records in the query table by pressing CTRL-A or by choosing Edit - Select All.

The result seen in the map is that all of the objects in the query table are selected. However, even though the selection mode is set to Select Replace the province of Oaxaca is still selected.
Next, we can click on the record handle for the Durango province to select it. Because the selection mode is Select Replace, selecting Durango deselects the other records in the query table.

We can see in the map that the province of Durango is selected while the other provinces in the query table are no longer selected.

Finally, with the focus on the query table window we can choose Edit - Select Inverse to invert the selection so that all records except Durango are selected.
In the map window we can see that Durango province has been deselected and all of the other provinces in the query table have been selected. As before, Oaxaca province is still selected.

**Lookup Values and CStr and CAST**

Invoking `CStr` on a lookup value or using `CAST` to convert the value to a string returns the descriptive name of the lookup value.

**Example**

We have a table `T` with a lookup column Region with values East and West. We want to select all records with the value of Region being East. We can do this with the following query:

```sql
SELECT * FROM [T] WHERE CStr([Region]) = "East";
```

**Example**

We have a drawing `D` and we want to select all areas it contains. We can do this with the following query:

```sql
SELECT * FROM [D] WHERE CAST([Type (I)] AS TEXT) = "Area";
```

**See Also**

Sample Queries  
Using SQL to Select Map Objects  
Spatial Extensions  
Geocoding Extensions  
Geometry in Tables
Queries and Geoms

**Geoms** are drawing metrics stored in tables from which linked drawings may be created. See the Linked Drawings and the Geometry in Tables topics for an introduction to geom data.

The numerous Spatial Extensions in Manifold SQL allow using SQL as a very powerful and flexible way of manipulating drawing objects when data for object metrics is stored as a geom in a table. In this conceptual model, all object data is stored as geoms in tables (or converted to geoms on the fly from an existing drawing) and whenever a drawing is desired it is created on the fly as a linked drawing.

In this model SQL becomes the user interface for manipulating data that is geographic or geometric in nature. Although it may seem retrograde to move away from a WYSIWYG, point and click visual interface into an SQL interface, for certain analyses or manipulations of data SQL offers a tool of formidable compactness and power that can be a better approach. This is especially true when automating procedures that are ill-suited for a lengthy series of point-and-click visual operations.

This topic provides a miscellaneous collection of notes and examples using spatial SQL with geoms to illustrate the wide-ranging power of spatial SQL and geoms.

### Create a Triangulation

Triangulate all points in drawing D and output triangulation lines:

```sql
SELECT TriangulationLines(AllCoords([Geom (I)])) FROM [D] WHERE IsPoint([ID])
```

### Using the SPLIT BY Clause

The SPLIT BY clause allows us to split tables using the content of one or more columns.

The SPLIT BY clause has the reverse effect of the GROUP BY clause: the GROUP BY clause collapses groups of records into single records, and the SPLIT BY clause explodes single records into groups of records. Unlike the GROUP BY clause, the SPLIT BY clause does not require the use of aggregate functions.

Consider a hypothetical table with columns called **ID** (an integer field) and **List** (a text field). The values in the **ID** column are used to uniquely identify each record. The values in the **List** column contain a list of names separated by commas. Let us say we want to transform this table so that each original record is replaced with one or more records with individual names in the **List** field corresponding to the same **ID**.

For a conceptual example, if we had a table...

<table>
<thead>
<tr>
<th>ID</th>
<th>List</th>
</tr>
</thead>
<tbody>
<tr>
<td>37</td>
<td>Orange, Blue, Red</td>
</tr>
<tr>
<td>38</td>
<td>Yellow, Green</td>
</tr>
<tr>
<td>39</td>
<td>Purple</td>
</tr>
</tbody>
</table>

...we would like to transform it into...

<table>
<thead>
<tr>
<th>ID</th>
<th>List</th>
</tr>
</thead>
<tbody>
<tr>
<td>37</td>
<td>Orange</td>
</tr>
<tr>
<td>37</td>
<td>Blue</td>
</tr>
<tr>
<td>37</td>
<td>Red</td>
</tr>
<tr>
<td>38</td>
<td>Yellow</td>
</tr>
</tbody>
</table>
We can use the SPLIT BY clause to accomplish the above. The above is a conceptual example only to illustrate how a group or list of items is split by a given field, the ID field. In actual practice, the SPLIT BY clause is used to split tables by one or more geometry columns. For example, we could use the SPLIT BY clause to split a table of lines by branches.

A split function is used to control how the SPLIT BY clause splits data. Currently, the SPLIT BY clause supports the following split functions:

- Branches: Splits geoms by branches
- Islands: Splits area geoms by islands
- Coords: Splits geoms by coords (inflection points).

### SPLIT BY Examples

The following query outputs each branch of each object in drawing D as a separate geom:

```
SELECT [ID], [P] FROM [D] SPLIT BY Branches(Geom([ID])) AS [P];
```

The following query creates a convex hull of each area in drawing D and outputs its coords (inflection points). Linking this query as a drawing will produce a point object at each coord of each created convex hull:

```
SELECT [ID], [P] FROM [D] WHERE IsArea([ID]) SPLIT BY Coords(ConvexHull(Geom([ID]))) AS [P];
```

The following query creates a Voronoi diagram for all points in drawing D and outputs the Voronoi areas for each point:

```
SELECT [P] FROM (SELECT Voronoi(AllCoords(Geom([ID]))) AS [V] FROM [D] WHERE IsPoint([ID])) SPLIT BY Branches([V]) AS [P];
```

The following query creates a Voronoi diagram for all points in drawing D and outputs the Voronoi areas for each point whose ID exceeds 5:

```
SELECT [ID], [P] FROM (SELECT [P] FROM (SELECT Voronoi(AllCoords(Geom([ID]))) AS [V] FROM [D] WHERE IsPoint([ID])) SPLIT BY Branches([V]) AS [P]) INNER JOIN [D] ON Contains([P], [ID]) WHERE IsPoint([ID]) AND [ID] > 5;
```

### The CoordSys and AssignCoordSys Functions

The CoordSys query function loads a coordinate system from a component or a preset with use as follows:

- `CoordSys("Latitude / Longitude")` loads a coordinate system from the Latitude / Longitude preset.
- `CoordSys("D" AS COMPONENT)` loads a coordinate system from a component named “D”.
- `CoordSys([D])` loads a coordinate system from a preset whose name resides in a column named “D”.
- `CoordSys([D] AS COMPONENT)` loads a coordinate system from a component the name of which resides in a column named “D”.

The AssignCoordSys query function assigns a coordinate system to a geom.

The following query selects each object in drawing D and interprets its coordinates as Latitude / Longitude coordinates.
SELECT AssignCoordSys(Geom([ID]), CoordSys("Latitude / Longitude")) FROM [D];

The Project Function

The Project query function projects a geom to a specified coordinate system.

The following query selects each object in drawing \textit{D} and projects it to the coordinate system used in drawing \textit{E}:

SELECT Project(Geom([ID]), CoordSys("E" AS COMPONENT)) FROM [D];

Height Functions and Surfaces

The Height, HeightMax, HeightMin, and HeightSum query functions return the average, maximum, minimum, and summed height of pixels in given surface covered by a geometry object.

The following query selects all points in drawing \textit{D} along with the heights at their respective locations in surface \textit{S}:

SELECT [ID], Height([S], Geom([ID])) FROM [D];

The following query selects all areas in drawing \textit{D} for which the maximum height in \textit{S} exceeds 1000:

SELECT [ID] FROM [D] WHERE HeightMax([S], Geom([ID])) > 1000;

The HeightSum function is used to compute cumulative figures such as population in a given region or to compute the volume of a surface (the HeightSum value multiplied by the size of a pixel).

Creating New Points

The NewPoint and NewPointLatLon query functions create point geoms in either the default (Orthographic) or Latitude / Longitude coordinate systems.

The following query uses the \textit{UTMX} and \textit{UTMY} columns in table \textit{T} to create points in the UTM Zone 17N coordinate systems:

SELECT AssignCoordSys(NewPoint([UTMX], [UTMY]), CoordSys("Universal Transverse Mercator - Zone 17 (N)")) FROM [T];

The following query selects all areas in drawing \textit{D} that contain the Latitude / Longitude location 89W 32N:

SELECT [ID] FROM [D] WHERE Contains(Geom([ID]), NewPointLatLon(-89, 32));

Linear Referencing

The LinePoint query function creates a point on a line geom at a given distance from the start of the line. The LinePart query function creates a new line constituting that part of a line geom between a given pair of distance limits.

Distances continue between branches: If a line contains two branches with lengths of 5 and 6, the first branch contains distances from 0 to 5 and the second branch contains distances from 5 to 11. Obviously, it is easy to get confused when using linear referencing with multi-branched lines so it make sense to split multi-branched objects into single-branched lines before using linear referencing.

Unless we specify a unit parameter, all distances are measured in native drawing units.
**Linear Referencing Example**

Let us say we have a drawing $D$ and a table $T$ which contains records in the form: $ID$, $F$, $T$, where $ID$ is the ID of a line in $D$, $F$ is a "from" distance on that line, and $T$ is a "to" distance on that line.

The following query traverses all records in $T$ and extracts those parts of lines it specifies:

```
SELECT LinePart([T].[ID], [F], [T]) FROM [T] INNER JOIN [D] ON [D].[ID] = [T].[ID];
```

The following query traverses all records in $T$ and extracts all "to" points it specifies:

```
SELECT LinePoint([T].[ID], [F]) FROM [T] INNER JOIN [D] ON [D].[ID] = [T].[ID];
```

**Queries and Coordinate Systems**

When queries are used to create objects in linked drawings the objects in those drawings will be based upon the coordinate system in use in whatever source drawings or geoms are used in the query. The units used to compute, say, a buffer will be the local units used in the projection.

Suppose we have a drawing containing point objects that is called $Points$. Consider a query like the following:

```
SELECT Buffer(Geom([ID]), 1000, "mi") FROM [Points];
```

Suppose a linked drawing is created from this query. The buffer objects the query creates may or may not appear to be circles depending upon the coordinate system (projection) used in the $Points$ drawing. The query will create buffer areas such that the outline of each area is the same distance, 1000 miles, from the central point for each area. However, this may or may not be shaped like a circle depending on the projection used.

If the $Points$ drawing contains one point and the projection for the $Points$ drawing is the Orthographic or similar azimuthal projection centered exactly on that one point then the result will be a true geographic circle. However, if the projection is otherwise what will appear on the screen will be a circular area (because in the drawing's local coordinate system each location on the boundary of the buffer will be equidistant from the point) but it will not actually draw a geographic circle on Earth.

For illustrations of this effect and the difference between drawing a circle upon a projected map and creating a geographic circle, see the Adding Shapes topic.

**See Also**

Linked Drawings
Geometry in Tables
Spatial Extensions
Queries and Images or Surfaces
Queries and Images or Surfaces

Queries may be written using the virtual tables of images or surfaces. Such queries can extract data from images or surfaces or they can manipulate the data in images or surfaces. See the Virtual Tables for Images and Surfaces topic for a list of columns in a virtual table and other details on virtual tables.

Queries can refer to the virtual table of an image or a surface by using the name of the image or surface. For example, the following query displays the total number of selected pixels in an image named **Bronze**:

```
SELECT Count(*) FROM [Bronze] WHERE [Selection (I)];
```

The following query displays the total number of selected pixels in a surface named **Foothills**, the heights of which do not exceed 100:

```
SELECT Count(*) FROM [Foothills] WHERE [Selection (I)] AND [Height (I)] <= 100;
```

**Example**

Consider the sample **bronze** image in which a rectangular region of pixels has been selected.

We can write a query, called **Bronze Query**, which selects selected pixels in the image.

```
SELECT * FROM [Bronze] WHERE [Selection (I)];
```

The above query uses the **Selection (I)** intrinsic field of the virtual table to select all fields for all pixels that are selected.

If desired, we can now create a linked image using that query. We choose File - Link - Image and in the Files of type box in the Link dialog we choose This Project().
In the Link Data dialog we choose the query we created. The rest of the fields use default values since we know the query has been created from an image using SELECT * so that all the required columns are generated by the query.

The result of the image linking operation is that a new component, a linked image, appears in the project pane.

If we open the linked image we see that it consists of the pixels that were selected in the image which is the subject of the query from which the linked image was created.
Suppose we change the selection in the **Bronze** image.

If we click on the linked image and do a **View - Refresh Data** command then the linked image will automatically change to show the current results of the query. Note that, since images must be rectangular in extent, any additional pixels required to create a rectangular image will be invisible.

**Example**

Consider a portion of the example Montara Mountain surface on the Manifold CD.
We can create a query that selects all pixels where the height of the surface is greater than or equal to 1000.

We shall call the query **Montara Query**. We can then create a linked surface based upon that query.

To do so, we choose **File - Link - Surface** and in the Data Source dialog we choose **This Project**. In the resulting **Link Surface** dialog, seen above, we choose the **Montara Query** query and use default settings otherwise.
The result is that a linked surface appears in the project pane. Note that the linked surface has a terrain associated with it just as would any other surface.

If we open the linked surface, we can see it contains those pixels in the original surface that are at a height of 1000 or greater. Note that the new surface is not colored with any palette. We can use the **View - Display Options** command to color it as we see fit. If we would like to duplicate the coloring used in the original Montara surface, we can open the Montara surface, choose **View - Display Options** and then use **Save to File** to save the exact palette being used to an .XML file.

We can then open the linked surface, choose **View - Display Options** and use **Load from File** to load the palette desired from the .XML file to which it was saved.

**Projections and the OPTIONS Clause**

By default, queries using virtual tables for images or selections do not save projection information. The image and surface created in the above examples will be created in a default Orthogonal projection that is not georeferenced. To save projection information, we can use the OPTIONS clause with the **CoordSys** query function. The OPTIONS clause allows a declaration of options for query execution and usage. At the present time, the only option for this clause is the **CoordSys** query function.

**Note:** The OPTIONS clause must be the first clause in the query, since it potentially specifies the operation of all the rest of the query.

Consider the following two queries operating on a surface named **Surface**:

```
SELECT * FROM [Surface] WHERE [Height (I)] > 100;

...and...

OPTIONS CoordSys("Surface" AS COMPONENT);
SELECT * FROM [Surface] WHERE [Height (I)] > 100;
```
Both queries select all pixels with a height greater than 100. However, linking a surface from the first query will create a surface in the default Orthographic projection, since pixels do not carry coordinate system data, but linking a surface from the second query will create a surface using the coordinate system (that is, the projection) of the original surface.

The OPTIONS clause works for drawings, too.

**Example**

We have a large surface and we would like to create a query that grabs only a portion of that surface within a given latitude and longitude range. We will then create a linked surface based on the query. The result will be a surface that is only a portion of the larger surface. We will use an OPTIONS clause in the query to make sure the resulting linked surface is correctly georegistered using the same projection as the original surface. This is a common technique for Internet Map Server (IMS) applications.

We begin with a surface showing dropout rates from high school in the US (blue represents lower dropout rates, red shows higher dropout rates). This is the same surface created in the Displaying Data in a Gradient Map example. We have colored the background black and have turned on a graticule showing latitude and longitude lines every five degrees.

We write a query called Surface Query that selects pixels between -85 and -80 degrees longitude and between 35 and 40 degrees latitude. The OPTIONS clause will ensure that linked surface created from the query will use the same projection as the original surface.
If we create a linked surface from the query we see that only that portion of the surface within the desired latitude and longitude ranges is in the linked surface. [The illustration seen above was created by using View - Display Console - Save to File from the original surface to save the palette, and then View - Display Console - Load from File in the linked surface to load the saved palette. This assures that both surfaces are colored the same way.]

For use in an application like an IMS website, we would probably rewrite the query used in this example as a parameter query as seen above. This would allow users to specify the minimum and maximum longitude and latitude range to get the desired portion of the surface.

To see the effect of parameter queries when working interactively with Manifold at the PC console (that is, not using IMS) we will have to manually refresh the linked component using the following procedure:

Create a parameter query. Launch the query using the ! button and supply values for the parameters. Next, link a surface from the query. The surface will use the data computed with the parameters that were supplied. Launch the query one more time and supply new values for the parameters. Refresh the surface by invoking View - Refresh Data. The surface will then use data computed with the new parameters.

**Example**

Spatial SQL can include functions for dynamic geometry that can be used to create linked drawings, that is, linked drawings that are created from spatial SQL queries. These can even be created from queries using an image's or a surface's virtual table.

Suppose we start with the surface from the previous example, with part of the surface selected as seen above.
We can write a query, called **Selection Points**, that creates a point for each selected pixel in the surface. The OPTIONS clause at the beginning of the query assures that any linked drawing created from this query will have the same coordinate system as the surface and thus will be correctly georegistered.

Using **File - Link - Drawing** with **This Project** in the Data Source dialog, we can create a linked drawing. The **Link Drawing** dialog is completed as seen above.

**Link Drawing**

- **Data source:** This Project
- **Type:** Table with geometry column
- **Source:** Selection Points
- **Columns:**
  - Column
  - Height (')
- **Geometry:** Column
- **Geometry type:** Geometry
- **Version:** [None]

The result is a linked drawing that consists of a mass of points arranged in the shape of the selected region of the surface.
The drawing is read-only, because it is created based on a query. If we want to use this drawing in subsequent operations and we would like it to be writable, we can make a writeable copy of it by right clicking on the linked drawing in the project pane and choosing Duplicate from the context menu. That will create a copy of the drawing as a separate, ordinary drawing.

Alternatively, we could right click on the linked drawing in the project pane and choose unlink to convert it from a linked drawing into a regular drawing. Choosing unlink sever all connections with the query.

Relink and Unlink

If a connection is lost between a linked image or surface and its originating data, the Image - Relink or Surface - Relink command allows us to restore the connection. If we would like to convert a linked image or surface into a regular image, the Image - Unlink or Surface - Unlink command will sever all connections to the originating source and will convert it into a regular image or surface.

Updates

Suppose we create a linked image or surface from a query or table. If the data changes we must use View - Refresh Data to update the linked image or surface. Simply closing and opening the component is not enough to refresh it.

Refreshing a component linked from a table or query is always done manually or via a script. One of the reasons for this is that the table or query providing data for the linked component may be external to Manifold. For example, it may reside in a remote database managed by SQL Server or Oracle. In such cases, depending upon the nature of the connection to that database, it could be very computationally expensive to determine when the remote data has been updated and thus a change in the linked component is required. Some queries can also be very expensive to run in terms of processor bandwidth and user interface pauses.

For these reasons, Manifold caches linked components so that the system is not loaded running unnecessary queries or continuously making computationally expensive determinations that no changes have occurred.

Units of Measure in Linked Surfaces

A limitation of Manifold's linked surfaces mechanism is that File - Link - Surface dialog expects heights to be represented in meters. If a linked surface is generated using a data set in which heights are expressed in feet or some other unit, then the hill shading effect will be overly dark.

A workaround for this limitation is to convert heights to meters if they are in some other unit. This may easily be done within the spatial query from which the linked surface is generated. For example, the Montara Mountain on the Manifold CD expresses heights in feet. To use heights in meters, we could write the following query:

```sql
OPTIONS CoordSys("Montara" AS COMPONENT);
SELECT [Height (I)]*0.3048 AS Z, [X (I)], [Y (I)] FROM Montara;
```

We could then use Z as the height in the Height box in the Link Data dialog.

Checking for Invisible Pixels

When writing a query that uses an image's or a surface's virtual table, it is usually wise to add a check for invisible pixels. Images or surfaces may have large regions of invisible pixels, and it usually does not make sense to return via a query many pixels that cannot be seen. We can easily accomplish this using the Invisible (I) field. For example, the Montara Mountain on the Manifold CD expresses heights in feet. To use heights in meters, we could write the following query:

```sql
OPTIONS CoordSys("Montara" AS COMPONENT);
SELECT [Height (I)]*0.3048 AS Z, [X (I)], [Y (I)] FROM Montara;
```

We could then use Z as the height in the Height box in the Link Data dialog.
Queries

* FROM [Surface]
WHERE
  ([Longitude (I)] >= MinLon AND [Longitude (I)] <= MaxLon) AND
  ([Latitude (I)] >= MinLat AND [Latitude (I)] <= MaxLat) AND
  NOT [Invisible (I)];

To illustrate that different indentation styles are possible the query text above uses a different style of indenting than the other examples in this topic. Manifold SQL is very tolerant of “white space” characters such as tabs, extra spaces or returns so that we can arrange our text using whatever style we think is most comprehensible.

Examples of UPDATE Queries

Because some fields of an image’s or surface’s virtual table are writeable (see the Virtual Tables for Images and Surfaces topic for a list of writeable fields), we can use UPDATE queries to change an image or surface.

Make all pixels with a value in the alpha channel greater than 200 invisible:

UPDATE [Image] SET [Invisible (I)] = True
WHERE [Alpha (I)] > 200;

Make all selected pixels red:

UPDATE [Image] SET [Color (I)] = RGB(255, 0, 0)
WHERE [Selection (I)];

Raise heights of all selected pixels by 10:

UPDATE [Surface] SET [Height (I)] = [Height (I)] + 10
WHERE [Selection (I)];

Select all pixels with heights between 10 and 30:

UPDATE [Surface] SET [Selection (I)] = True
WHERE [Height (I)] BETWEEN 10 AND 30;

Select all pixels with heights between 10 and 30 degrees and deselect all other pixels:

UPDATE [Surface] SET [Selection (I)] =
  ([Height (I)] BETWEEN 10 AND 30);

Examples of Analysis done with Queries

Queries may be used to perform analysis on images and surfaces as the following examples illustrate.

Determine the number of pixels falling into a given height range:

SELECT Count(*) FROM [Montara]
WHERE [Height (I)] BETWEEN 1000 AND 1200;

Determine the number of pixels falling into a given intensity range:

SELECT Count(*) FROM [Montara]
WHERE Intensity([Color (I)]) BETWEEN 240 AND 255;

Create a table of all heights along with their relative frequencies:

```
SELECT
  Sum(Iif([Height (I)] <= 499, 1, 0)) / Count(*),
  Sum(Iif([Height (I)] BETWEEN 500 AND 999, 1, 0)) / Count(*),
  Sum(Iif([Height (I)] BETWEEN 1000 AND 1499, 1, 0)) / Count(*),
  Sum(Iif([Height (I)] >= 1500, 1, 0)) / Count(*)
INTO [Percentages] FROM [Montara];
```

**Tech Tip**

Consider the linked drawing created from a surface in an example above, which contains a mass of points arranged in the shape of the selected region of the surface. How might we use those points to create an area object in the same shape?

This is a surprisingly complex task due to the ambiguities involved in determining where "holes" or "islands" should be placed in the resultant area object. It's an easy thing for the human eye to see in cases of such dense point placement, but in regions where points might be very sparsely placed it could be difficult even for the human eye to decide where an area should begin or end.

For an automated process, we can create buffers or rectangular areas or a convex hull about the points to approximate the region covered with an area object. The following queries provide some example approaches.

**Buffers (slowest, area slightly increased):**

```
SELECT
  UnionAll(Buffer(AssignCoordSys(
    NewPoint([X (I)], [Y (I)]),
    CoordSys("Montara" AS COMPONENT)), 10, "m"))
FROM [Montara] WHERE [Height (I)] > 1600;
```

**Rectangles (slightly faster than buffers, area slightly increased):**

```
SELECT
  UnionAll(BoundingBox(Buffer(AssignCoordSys(
    NewPoint([X (I)], [Y (I)]),
    CoordSys("Montara" AS COMPONENT)), 10, "m")))
FROM [Montara] WHERE [Height (I)] > 1600;
```

**Convex hull (fastest, but probably least "accurate"):**

```
SELECT
  ConvexHull(AllCoords(AssignCoordSys(
    NewPoint([X (I)], [Y (I)]),
    CoordSys("Montara" AS COMPONENT))))
FROM [Montara] WHERE [Height (I)] > 1600;
```

**Notes**

Many of the above techniques (such as using a spatial query to select by latitude and longitude) can be used with images, surfaces or, with slight modifications, drawings.

**See Also**
Linked Images
Linked Surfaces
Virtual Tables for Images and Surfaces
Queries and Geoms
Linked Drawings
Geometry in Tables
Spatial Extensions
Geocoding

Geocoding is the process of finding the correct geographic position for a record in a table and then adding latitude and longitude values for each record in a table. Before continuing with this topic, please first read the About Geocoding topic.

There are two main types of geocoding:

- **Street Address Geocoding** uses the values of address fields such as street, city, state or province, and postal code to estimate the location for that record. Street address geocoding requires a specialized database that gives the position of various address ranges for different streets. As a practical matter, address information is rare and expensive outside of the United States, although the auxiliary use of MapPoint for Canada, Mexico and eleven countries in Europe provides a low cost option in those regions. Even within the United States, because of the highly irregular nature of addresses in the US and the poor quality of most address databases, street address geocoding provides at best an approximation of where a given address is truly located. See the Street Address Geocoding topic for more information.

- **Spatial Geocoding** matches the values of records in a database table with field values in the table of an available drawing. The spatial layout and locations of the drawing objects are used to determine locations for records that match object data fields. In Manifold marketing documents this capability is at times referred to as “generic geocoding.” Spatial geocoding has the advantage of allowing spatial matches to any available drawing. See the Spatial Geocoding with Match topic for information on spatial geocoding with Manifold.

Requirements

**Spatial geocoding** using **Match** does not require purchase of any options, but it does require a drawing to which matches can be made. For example, if we would like to geocode a table by Chilean postal codes then we will need a drawing showing centroids of Chilean postal codes that **Match** can use for the geocoding process.

Street Address Geocoding has two requirements:

- We must acquire and install a license for the optional Manifold Geocoding Tools package. The **Geocoding Tools** package is a built-in part of Manifold **Universal Edition**, but it is an optional extension to other Manifold System editions.

- We must have access to one or more of the geocoding data sources that can be used by the internal Manifold geocoding engine. Once enabled by installation of the **Geocoding Tools** package, Manifold can use a variety of geocoding data sources, including the Manifold Geocoding Database for US streets provided as a free download on the **manifold.net** site, MapPoint North American or European editions, user-provided geocoding data extensions or Manifold Geocoding Servers accessed through the web. The geocoding data source in use must be appropriate to the geographic region of interest. For example, if we have the US street geocoding database provided as a free download on the **manifold.net** site, that will allow us to geocode street addresses in the US but not in Europe.

See Also

- About Geocoding
- **Geocoding Tools**
- **Geocoding Data Sources**
- Create a Map from a Geocoded Table
- Create a Geocoded Table from a Map
- Create a Linked Drawing from a Geocoded Table
- Street Address Geocoding
- **Geocoding with MapPoint**
- **Geocoding Data Extensions**
- Manifold Geocoding Servers
- Spatial Geocoding with Match
Geocoding Tools

The Geocoding Tools package is an optional extension to Manifold System that provides street address geocoding capability as well as access to the Manifold Geocoding Database data provided as a free download on the manifold.net site for both US street address geocoding as well as the easy creation of drawings showing US streets. The Geocoding Tools package also requires access to one or more of the geocoding data sources that are required for street address geocoding.

If you do not have the optional Geocoding Tools package you will not be able to use street address geocoding functions within Manifold (regardless of the data source used) or take advantage of the street address geocoding data provided as a free download on the manifold.net site for the United States. If you do not also have at least one geocoding data source appropriate to the geographic region of interest (regardless of whether you have installed the Geocoding Tools package), you will not be able to use street address geocoding functions. If you plan on using the Manifold Geocoding Database see the Geocoding Data Sources topic for instructions on installing the database provided as a free download on the manifold.net site.

Very important: the Geocoding Tools package will not be operational until you install at least one geocoding data source. You must install at least one geocoding data source.

The Geocoding Tools extension adds functionality in several areas:

- Street address geocoding within the United States using the Manifold Geocoding Database provided as a free download on the manifold.net site.
- Street address geocoding within North America or Europe using the appropriate edition of Microsoft’s MapPoint product as a data source.
- Street address geocoding anywhere in the world using user-provided geocoding data extensions in either range or points of interest formats.
- Street address geocoding using web-based Manifold geocoding servers to provide geocoding services.
- Easy import of drawings showing US streets from the Manifold Geocoding Database provided as a free download on the manifold.net site as set forth in the Import Drawing - Geocoding Database topic.
- Geocoding Extensions may be used within SQL to allow use of street address geocoding functions within SQL queries.
- Programming objects such as the Geocoder object may be used from scripts or other programs or from within IMS applications.
- Street addresses and ZIP codes may be used in the Edit - Go To dialog.

If you do not have the optional Geocoding Tools package enabled you will not have the above capabilities enabled within Manifold System. This documentation describes these commands but they will not be enabled if you do not have the Geocoding Tools package enabled.

Installing Geocoding Tools

The Geocoding Tools package is built into your Manifold System installation and becomes enabled for use when activated. Activate it by providing a Geocoding Tools serial number and Activation key that turn on Geocoding Tools functionality. The process of activating the Geocoding Tools extension is very similar to that used to activate Manifold System. See the Activation Keys and Serial Numbers topic and the Installing and Activating a Manifold Extension topic for details.

Some Manifold product options, such as Universal Edition or Ultimate Edition, use a single serial number and Activation key to enable permanent installation of both Manifold System as well as extensions. If you have activated Universal Edition or Ultimate Edition you do not need to individually turn on Business Tools or Geocoding Tools or Surface Tools. Activating Universal Edition or Ultimate Edition will also automatically turn on all three extensions.

Turning on Geocoding Tools

If you have licensed Universal Edition, you do not need to turn on Geocoding Tools. The Geocoding Tools extension is automatically enabled with a Universal Edition serial number.

If you have not licensed Universal Edition and you wish to add Geocoding Tools functionality to your installation, begin by acquiring a Geocoding Tools license from manifold.net, which will provide you with a Geocoding Tools serial number.

2. Launch Manifold System. Close any projects that may be open.

3. Choose Help - Activate Extension

4. Enter the Geocoding Tools serial number. Enter it exactly as it was issued to you by manifold.net. Do not change upper case to lower case. Do not replace hyphens with space characters or make any other changes to the serial number. If desired, you may also enter an Activation key for the serial number as well. Press Accept.

5. Manifold will then exit. When re-launched, the Geocoding Tools package will be enabled.

6. Using the serial number alone the Geocoding Tools package may be run for 30 days from the date the serial number was issued. After that, an Activation Key must be fetched and provided to the Help - Activate Extension dialog together with the serial number. Once an Activation Key and a serial number have been provided to the Help - Activate Extension dialog the extension will be permanently installed. Don't forget to login as Administrator when permanently activating the Geocoding Tools extension.

7. Install a geocoding data source - You must have at least one geocoding data source installed to operate the Geocoding Tools package. Most US users will download and install the Manifold Geocoding Database provided as a free download on the manifold.net site

---

**Status of Extensions**

The Help - About dialog shows all extensions that have been installed. If a serial number only has been used to install an extension, the number of days left before an Activation key is also required will be shown.

**See Also**

Geocoding Data Sources  
Street Address Geocoding  
Geocoding with MapPoint  
Geocoding Data Extensions  
Manifold Geocoding Servers
Street Address Geocoding

Street address geocoding is the process of determining an estimated latitude and longitude position for the location of a street address, such as "1600 Pennsylvania Avenue NW, Washington, DC 20500." Manifold includes a sophisticated street address geocoding engine, but it must be enabled through the installation of Geocoding Tools and supported by access to an appropriate geocoding data source. Before continuing with this topic, please first read the About Geocoding topic.

Street Address Geocoding within Manifold is mostly used in two situations:

- Adding a latitude and longitude location to each record in a table of addresses. The table then becomes a geocoded table and can easily be copied and pasted as a drawing. Geocoding the street addresses allows us to see where they are located as points in a drawing.
- Using a street address within interactive commands such as the GoTo command. In that case the street address is "geocoded on the fly" to determine the latitude and longitude location to be used by the command.

In addition to the above uses, Manifold street address geocoding facilities are also used within SQL queries and are used by programmers via the Manifold API.

Required Geocoding Tools

Manifold's street address geocoding capability becomes operational only when the optional Manifold Geocoding Tools package is installed. If you do not have the optional Geocoding Tools package, you will not be able to use street address geocoding functions within Manifold (regardless of the data source used) or take advantage of the Manifold Geocoding Database provided as a free download on the manifold.net site for the United States.

Note that Manifold Universal Edition automatically installs the Geocoding Tools package. If you are using Universal Edition, you do not need to perform an extra Geocoding Tools installation as it has already been installed.

Required Geocoding Data Sources

In addition to installation of Geocoding Tools, the Manifold installation must have access to a geocoding data source. Once enabled by installation of the Geocoding Tools package, Manifold can use a variety of geocoding data sources, including the Manifold Geocoding Database for US streets provided as a free download on the manifold.net site, MapPoint North American or European editions, user-provided geocoding data extensions or Manifold Geocoding Servers accessed through the web.

Without a geocoding data source such as the US streets Manifold geocoding database or MapPoint, Manifold's street address geocoding commands cannot function. The Manifold Geocoding Database supports US street address geocoding only. The Microsoft MapPoint North American edition product will allow street address geocoding in the US and Canada, while the European edition of MapPoint will allow street address geocoding in eleven European countries. User-supplied geocoding extensions can cover whatever region the user desires. Web-based servers vary greatly in the regions they cover. See the Geocoding Data Sources topic for information on these sources as well as information on installing the Manifold Geocoding Database provided as a free download on the manifold.net site.

Limitations of Street Address Geocoding

Unless a special type of geocoding data source is used that provides data in "points of interest" format, the latitude and longitude location created for a given address by Manifold is only an estimate. In all cases, the accuracy of either the estimate or a putatively-precise location made possible by points of interest data sources will be determined by the accuracy and detail of the geocoding data source used. As the famous saying has it, "Garbage in, garbage out."

If the data source in use does not include a particular street or does not include information on the address of interest, that address cannot be located. At times addresses in use in the US and other countries will vary considerably from the usual pattern to the degree they cannot be easily cited within a data source.

For these and other technical reasons, street address geocoding normally cannot find latitude and longitude locations for all street addresses submitted and the locations which are found will at best be estimates and not exact locations. Street address geocoding using high quality geocoding data sources in urban areas that feature
regular address patterns on streets will often result in high hit rates, with over 90% of addresses routinely being located and geocoded to within a few dozen meters of their actual locations. Street address geocoding done in rural areas with irregular address patterns can have abysmally poor hit rates and accuracy unless the geocoding data source is a user-supplied geocoding data extension using points of interest format.

Requirements for Records to be Geocoded

Manifold can geocode an entire table containing street address records, a process often called "batch street address geocoding" in GIS nomenclature. Records that are to be geocoded must have the following characteristics:

- **Standard fields** - Each record must include four standard text columns containing street address, city, state and ZIP code. Using Standardize, it is usually possible to fill in missing ZIP codes based on street address, city and state information. The Country field by default uses the United States; however, when geocoding outside of the United States using MapPoint the Country field should be used to specify the country for that address.

- **Well-formed addresses** - Address fields cannot contain multi-line values or special characters such as newlines.

- **Complete street addresses** - Street address fields should contain the true name of the street. For example, if an address is located on "Johnston Road" the street name should not just be something like "265 Johnston" but should include the full "265 Johnston Road" or include a standard abbreviation for "Road".

- **No extensions** - The column containing the street address must contain the street address only without any additional address information such as apartment number or other embellishments (such additional information can be kept in a separate field).

- **Text columns** - Columns used for geocoding must be text columns, not numeric columns. Note that many postal codes, such as ZIP codes, have leading "0" characters (with ZIP codes such as "02138") that are significant and will be preserved if a text column is used and lost if a numeric column is used.

- **International addresses** - See the Geocoding with MapPoint topic for additional requirements and limitations when geocoding Canadian or European street addresses using MapPoint.

When geocoding addresses, most effort will be expended on cleaning addresses and converting those that are in non-standard form into standard form. This is accomplished with the Standardize command. Once addresses have been standardized, we can use the Geocode command to geocode them.

The Manifold Geocoding Database or some other geocoding data source must be installed on for either of these commands to function. If a usable geocoding data source has not been installed, neither of these two commands will be enabled.

The Standardize and Geocode dialogs are related in how they use the geocoding database and both dialogs use the same Unmatched Records dialog to allow manual adjustments to records that cannot be automatically standardized or geocoded. The Standardize dialog uses the geocoding database to look up valid zip codes, city names, street names and address ranges. It can use that information to identify errors in records and to offer alternatives. The Geocode dialog uses presumably valid address information to determine, if possible, the latitude and longitude location of a specific address.

To geocode a table of street addresses:

1. Open the table of addresses. If the addresses are not known to be absolutely clean and well formed we should first run Table - Address - Standardize to eliminate common problems.
2. Choose Table - Address - Geocode. Specify the fields that are to be used for the standard geocoding fields of Address, City, State and Zip. Specify which fields are to contain the created latitude and longitude values or let Manifold create new columns for these values. Press OK.
3. Any records that cannot be automatically located in the geocoding database will be presented in the Unmatched Records dialog. This dialog will allow us to edit the record to make corrections. It will present a menu of nearest matches, if any, from which we can quickly choose a desired match.

After geocoding the table will contain latitude and longitude columns that will contain the latitude and longitude location of each record.

Standardize
The **Standardize** command is used to convert addresses from non-standard forms into the standard form expected by the Manifold geocoder. It has several functions:

- **Extract Address, City, State and Zip fields** - Given one or more address fields that contain a mix of street address, city, state and Zip code information, the command will attempt to parse the data and (usually) extract information into four newly-created standard **Address**, **City**, **State** and **Zip** fields. The command can normally extract the main street address from a single address field. For example, if we have a single database field called "Street Address" that contains a string such as "408 Queen Street Suite 35, Causeway Bay, Kansas" Manifold can parse the string to obtain "408 Queen Street" for the **Address** field.

- **Find missing Zip codes** - If a Zip code is not present the command can usually find the missing Zip code or present possible matches for approval.

- **Normalize address nomenclature** - If the **Normalize addresses** box is checked the command will convert address nomenclature into standardized abbreviations, such as "Rd" for "Road," and will convert state names into official postal abbreviations. The command will also correct elementary errors in case, such as converting "1208 Hennessy ROad" to "1208 Hennessy Rd".

- **Check for unknown locations** - The command will automatically check for impossible combinations of streets, towns and zip codes, including unknown town names, street names and other clearly inaccurate address data.

The **Standardize** command will process a table and then raise the **Unmatched Records** dialog to allow editing and selection of proposed matches the system thinks are close to what is intended.

### Standardize Addresses Dialog Controls

<table>
<thead>
<tr>
<th><strong>generate ... using</strong></th>
<th>Specify the columns to be used for standardized address fields and from which column each standardized address should be extracted. Either new columns may be automatically created or existing columns may be used.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Address</strong></td>
<td>Column to be used for the standard field that contains the building number and street name. Choosing [New Column] will add a new column to the table called <strong>Address</strong>.</td>
</tr>
<tr>
<td><strong>City</strong></td>
<td>Column to be used for the standard field that contains the city name. Choosing [New Column] will add a new column to the table called <strong>City</strong>.</td>
</tr>
<tr>
<td><strong>State</strong></td>
<td>Column to be used for the standard field that contains the state name. Choosing [New Column] will add a new column to the table called <strong>State</strong>.</td>
</tr>
<tr>
<td><strong>Zip</strong></td>
<td>Column to be used for the standard field that contains the five-digit postal ZIP code. Choosing [New Column] will add a new column to the table called <strong>Zip</strong>.</td>
</tr>
<tr>
<td><strong>Country</strong></td>
<td>Optional column for country name. The Manifold Geocoding engine recognizes typical variations of country names (&quot;US,&quot; &quot;USA,&quot; &quot;United States,&quot; &quot;The United States of America,&quot; and so on). Manifold will use the default country specified in the Tools - Options dialog for <strong>Standardize</strong> and <strong>Geocode</strong> operations if there is no <strong>Country</strong> field.</td>
</tr>
<tr>
<td><strong>Status</strong></td>
<td>If desired, add a column that reports the status of a particular record after processing. [None] does not report status. [New Column] will add a new column to the table called <strong>Status</strong>.</td>
</tr>
<tr>
<td><strong>Fail on</strong></td>
<td>Criterion to be used to declare a particular record to be unmatched.</td>
</tr>
<tr>
<td><strong>Normalize addresses</strong></td>
<td>Convert address nomenclature into standardized abbreviations, such as &quot;Rd&quot; for &quot;Road,&quot; state names into official postal abbreviations, etc.</td>
</tr>
<tr>
<td><strong>Skip completed</strong></td>
<td>Do not process records that already have values in the</td>
</tr>
</tbody>
</table>
records designated Address, City, State and Zip fields.

Fail on Options

any error Does not process record unless it is a perfect match in all fields. Equivalent to setting an "unknown building" error.

unknown street name, possible misspelling Matches to Zip and City and does not process record unless the street name is an exact match.

unknown street name, no similar names Matches to Zip and City, and attempts to match to similar street names if an exact match to the street name cannot be found. Uses “Soundex” and similar algorithms to attempt to find the right street in case of a street name misspelling. This is the default setting.

unknown zip Do not process record if the zip code cannot be found.

critical error Stop processing only on problems with an incomplete geocoding database or on hardware failure.

Status Values

When reporting the status of a record's processing in a Status field the following values are used, listed in order of increasing severity:

ok No errors in processing record.

unknown building The geocoder was able to locate the zip code and the street name, but the street does not appear to contain an address of the given number.

unknown street name, possible misspelling The geocoder was able to locate the zip code but not the street name; however, the zip code does contain some streets with names similar to the street name.

unknown street name, no similar names The geocoder was able to locate the zip code but not the street name. The zip code does contain any streets with names similar to the street name.

unknown zip code The geocoder could not locate the zip code.

critical error The geocoder could not function due to an incomplete geocoding database, a problem accessing the database or a hardware failure.

Search Priority

The biggest difficulties encountered when standardizing addresses arise from errors in the address. Addresses may be incomplete, missing zip codes, for example, or they may incorporate typographical or other errors that result in erroneous zip codes, city names, street names and building numbers. At times, cities may be called by local names that are different than those officially recorded by the US government. For example, a small town located North of Boston is called "Manchester," "Manchester-by-the-Sea" and "Manchester by the Sea".

Manifold helps deal with such errors by prioritizing searches for standardization and geocoding using the following order of precedence:

- Search by zip code and then search by street name and building number.
- If the zip code and street can be located but there is no address range on the street that allows the given building number the system will return the data found and will generate a unknown building error.
• If the zip code can be located but no such street name can be found within that zip code, the system tries to locate streets with similar names. Variations will be scanned for different prefixes (for example, "East" or "West" in combination with the street name), different types ("Road," "Street," "Drive," "Lane," etc.), typical abbreviations and possible misspellings. Finally, the system will search for street names that sound like the one given. If similar street names can be found, the system will return the data found and will generate an unknown street name, possible misspelling error.

• If the zip code cannot be found or if there are no streets with similar names found, the system tries to located street names (first, by exact matches and then approximate matches) in a wider geographic area, first within the city name given and then within the state name given. It then returns the data found and generates either an unknown street name, no similar names error or unknown zip error depending on whether the search started with a valid zip code or not.

The setting used for the Fail on parameter allow Manifold to accept failed address matches up to a given level of severity. For example, the default setting of unknown street name, no similar names for the Fail on parameter will allow Manifold to automatically accept records for which an unknown building or an unknown street name, possible misspelling error would be reported.

**Geocode**

The Geocode command takes an address in standard form, finds it in the Manifold Geocoding Database and then produces latitude and longitude values for the location of the address. Addresses to be geocoded must include the four standard Address, City, State and Zip fields.

Records that cannot be located within the error level specified by the Fail on parameter will be reported in an Unmatched Records dialog to allow editing and selection of proposed matches the system determines may be the correct locations. If a particular address cannot be located on a street, the system will report "Building not found" and will offer a match to the address. Choosing that match will choose a location for the address that is at the midpoint of the street segment for the street of that name.

Records that cannot be geocoded at all, even after manual intervention, will have zero values for latitude and longitude. When the table is copied and pasted as a drawing these records will be ignored if the Skip zero latitude / longitude records option is checked (the default setting) in the Paste As Drawing dialog.

If an address cannot be found in the available street address ranges for a specific street the command will choose a point near the middle of the street segment.

**Geocode Addresses Dialog Controls**

- **generate ... using** Specify the columns to be used for Longitude and Latitude, Status and Match type fields, and which columns in the table should be used for the standardized Address, City, State, and Zip fields.

- **Longitude** Table column to be used for the longitude of geocoded record. Contains 0 if the record was not geocoded. Use [New Column] to automatically create a new column called Longitude.

- **Latitude** Table column to be used for the latitude of geocoded record. Contains 0 if the record was not geocoded. Use [New Column] to automatically create a new column called Latitude.

- **Status** If desired, add a column that reports the status of a particular record after processing. [None] does not report status, [New Column] will add a new column to the table called Status.

- **Match type** If desired, add a column that reports the geocoding match type of a particular record after processing. [None] does not report match type, [New Column] will add a new column to the table called Match type.

- **Fail on** Criterion to be used to declare a particular record to be unmatched.
Offset Locations by
Offset the location for each geocoded record by the given number of units from the street line, to the left or right side of the street based on whether the address was found in the left or right range.

Skip completed records
Do not process records that already have values in the designated Address, City, State and Zip fields.

Fail on Options

any error
Does not process record unless it is a perfect match in all fields. Equivalent to setting an "unknown building" error.

unknown street name, possible misspelling
Matches to Zip and City and does not process record unless the street name is an exact match.

unknown street name, no similar names
Matches to Zip and City, and attempts to match to similar street names if an exact match to the street name cannot be found. Uses "Soundex" and similar algorithms to attempt to find the right street in case of a street name misspelling. This is the default setting.

unknown zip
Do not process record if the zip code cannot be found.

critical error
Stop processing only on problems with an incomplete geocoding database or on hardware failure.

Match type Values

When reporting the status of a record's processing in a Match Type field the following values are used, listed in order from most precise to least precise:

building
The zip code, street name and building number were used to identify a location at the building's number.

street
The zip code and acceptable street name were found but not the building's number so the location has been placed at the center of the bounding box for the street.

zip
Only a zip code was used so the location has been placed at the center of the bounding box for the zip code.

city
Only the city name was used so the location has been placed at the center of the bounding box for the zip code.

(empty string)
The record could not be geocoded and values of 0 have been written into both Latitude and Longitude.

Selections

Both the Standardize and the Geocode commands are auto-scoped: if a selection is present in the table they will operate only on the selected records.

Example
Suppose we have a table like that above, which contains two text fields: a Name field and a Street Address field. The table lists a few sushi restaurants near the USGS facility in Menlo Park, California. To geocode this table we must first standardize the addresses it contains using the Standardize command by choosing Table - Address - Standardize.

The Standardize Addresses dialog allows us to designate which fields will be used as sources to generate the five standard Address, City, State, Zip and Country fields and which existing fields, if desired, will host the four standard fields. We choose Street Address as the source for the new fields and [New Column] for each new field to be generated, so that a new column is created for each. Note: We will have to explicitly choose [New Column] for the Address column, since Manifold will by default the Street Address column into that field since it has "Address" in the name of the field and Manifold will guess that we might want to use it.
The result is that five new columns, Address, City, State, Zip and Country, have been created in the table and the relevant parts for each record have been extracted from the Street Address column and placed in the new columns.

Let's hide the Street Address column to reduce the size of the illustrations. We also hide the Country field since we don't need an explicit reminder that these locations are in the United States. That makes the table smaller but the Address and City fields look ugly all in capital letters. We can convert them to title case by using the Make Title Case operator in the transform toolbar for the Address and City columns.
That's better. Manifold will happily geocode whether or not columns contain all upper case letters so the above step is not necessary. But using title case makes the table more legible to humans so that's what we prefer to use. The table above is now in standard form so we can geocode it using Table - Address - Geocode.

The Geocode Addresses dialog allows us to choose which columns will be used for the four standard fields used by the Manifold geocoder. Even though the Country column is not visible, Manifold knows it is still in the table and can use it. Even if we did not have an explicit Country column Manifold would know to use the default country set in the Tools - Options dialog. We can also choose which columns will receive the generated latitude and longitude locations.

The Offset locations by checkbox will automatically position geocoded points to the left or right side of a street based on which side of the street that address falls on (according to the geocoding database). The distance and units boxes allow us to choose how far from the street centerline the geocoded points will be offset. The default offset is 50 feet.

The Skip completed records checkbox tells the dialog to ignore records that already contain data in the target fields.

If all records can be matched by the geocoder the result will be the addition of two new columns to the table called Latitude and Longitude that contain the latitude and longitude locations for each address. The table is now a geocoded table and can be copied and pasted as a drawing.
If we copy the table and paste it as a drawing, we can drag and drop the drawing into a map created from a drawing of roads in the Menlo Park and Palo Alto area. The points have been formatted as bright green dots.

**The Unmatched Records Dialog**

The Unmatched Records dialog is used with both the Standardize and the Geocode commands. In both cases, records that cannot be correctly standardized or geocoded are presented one by one within the Unmatched Records dialog to allow users to deal with each record on a case-by-case basis.

The dialog presents the address record being processed in editable boxes for Address, City, State and Zip fields. A list of possible alternatives found in the geocoding database is presented in a Found pane. We can edit the fields manually or, if we see a match that we like in the Found pane we can click on it to highlight the match and load it into the edit boxes in the dialog. Pressing Accept will save the values from the edit boxes into the record in the table, including the latitude and longitude in the case of the Geocode command.

If desired (say in the case of an obvious typographical error in the name of a street) we can edit the values in the Address, City, State and Zip edit boxes and then press Look Up to direct the system to check the new values against the geocoding database to see if they can be located. If the location is found we can press Accept to accept the geocoding for the edited address and to write the edited address values back into the table for that record. If we would like to go back to the previous set of values, before editing, we can press the Previous button.

If we would like to skip this record and continue with other unmatched records, we press the Skip button.

**Unmatched Record Dialog Controls**

- **Record**  Current record as it has been read from the table. A read-only edit box that can be used with Copy.
- **Address**  Address value now in use.
- **City**  City value now in use.
- **State**  State value now in use.
- **Zip**  Zip value now in use.
- **(result)**  Result of last Look Up operation.
- **Found pane**  A list of possible matches found in the geocoding database. Clicking on one of the matches will load it into the edit boxes. Double-clicking on one of the matches will load it and immediately Accept it as well and move on to the next record.
- **Accept**  Accept the current values in the edit boxes. Click on one of the choices in the Found pane and then choose Accept. This updates the table with specified values, replacing the current values of the record.
- **Look Up**  Look up the current address in the edit boxes in the geocoding database. When used with Standardize, this checks the possible validity of the address. When used
with Geocode, this looks up the location of the address.

**Previous**  Reload the previous values for this address into the edit box. Enabled if the address has been edited, either by choosing a possible match in the Found pane or by manually editing the address.

**Skip** Skips this record and continue with other unmatched records.

**Close** Skips this and all remaining unmatched records. Note that any Accept commands that have been issued for any unmatched records before a Close command will have already updated the table. Close is not the same as a Cancel of all changes made in this session with the Unmatched Records dialog.

The **Unmatched Records** dialog has keyboard accelerators for the Accept, Look Up, Previous, Skip and Close commands bound to ALT E, L, P, K and O keys. Pressing ALT-E is the same as clicking Accept. Using keyboard accelerators can help deal with very long lists of unmatched records.

### Skipping Unmatched Records when Creating Drawings

The results of the Geocode command are latitude and longitude values for each record that was successfully geocoded. Unmatched records that were skipped will have 0 values for their latitudes and longitudes. When copying a table that contains unmatched records and pasting it as a drawing, make sure the Skip zero latitude / longitude records checkbox is checked in the Paste As Drawing dialog. This will make sure that the unmatched records having 0 values for their latitudes and longitudes are not pasted as a cluster of points off the coast of Africa.

### Geocoding SQL Extensions

Manifold SQL includes geocoding extensions that operate with Manifold’s geocoding engine to perform spatial operations based upon an address string or zip code. Geocoding extensions will not work unless the US streets geocoding database is correctly installed. Geocoding extensions will not work with Manifold IMS unless the US streets geocoding database is installed within the Manifold application installation folder (normally, C:\Program Files\Manifold System). Therefore, the US streets geocoding database should be installed in the Manifold application installation folder on machines on which Manifold IMS operates.

**Boolean CloseToAddress** (Number ID, String Address, Number Distance, [String Unit])

Given an object ID, an address string, a distance and an optional distance unit determine if the object lies within the specified distance of the address.

**Boolean CloseToZip** (Number ID, String Zip, Number Distance, [String Unit])

Given an object ID, a ZIP code string, a distance and an optional distance unit determine if the object lies within the specified distance of the zip code centroid.

**Number DistanceToAddress** (Number ID, String Address, [String Unit])

Given an object ID, an address string, and an optional distance unit computes the distance between the object and the address.

**Number DistanceToZip** (Number ID, String Zip, [String Unit])

Given an object ID, a ZIP code string, and an optional distance unit computes the distance between the object and the zip code centroid.

### Notes on usage:
If an object is a line or an area the object's centroid is used for distance calculations.

If an address string produces more than one match, the system automatically selects the closest of the building-level matches (possibly with an "unknown street name, possible misspelling" error).

If an address string produces no building-level matches, `CloseToAddress` returns `False` and `DistanceToAddress` returns `-1`.

If the optional distance unit is omitted, the system will use the native measurement unit of the drawing or meters if the drawing is not projected.

Distances are great circle distances computed over a WGS84 ellipsoid and are accurate to 1 meter.

The geocoding functions cache returned geocoding data between subsequent calls.

Functions can be used from IMS as long as the geocoding database is located within the Manifold application installation folder (usually `C:\Program Files\Manifold System`).

### Geocoding Function Examples

```sql
SELECT * FROM Dealers
WHERE CloseToAddress(ID, "330 Lytton Ave, Palo Alto, CA, 94301", 10, "mi")
```

```sql
SELECT * FROM Dealers
WHERE DistanceToAddress(ID, "330 Lytton Ave, Palo Alto, CA, 94301", "mi") <= 10
```

```sql
SELECT * FROM Dealers
WHERE CloseToZip(ID, "94301", 10, "mi")
```

```sql
SELECT * FROM Dealers
WHERE DistanceToZip(ID, "94301", "mi") <= 10
```

All four examples have a similar function. The first query selects all objects in `Dealers` that are within 10 miles of the given address using the `CloseToAddress` function, while the second example performs the same task using the `DistanceToAddress` function. The third and fourth examples perform the same functions using the 94301 ZIP code.

To keep the user interface simple and to avoid the complication of dealing with possible user errors when entering address information into forms, many web applications with IMS will use the `CloseToZip` or `DistanceToZip` functions since these require the user to merely enter the ZIP code correctly. For many applications, such as locating a dealer, finding the closest objects to the ZIP centroid provides acceptable accuracy.

See the Units topic for a list of unit abbreviations that may be used to specify optional distance units.

### GoTo Extensions

When the US street address Manifold Geocoding Database is installed, the Edit - GoTo command will allow a GoTo to an Address or to a Zip code within the US. The address may be a full street address, or it may be a partial address, such as "Atlanta" or "Atlanta, GA" or "GA." Manifold includes a gazetteer of large city names so that "Atlanta" will find the large city in Georgia and not one of the various small towns of that name throughout the US. City names will take priority over states when spelled out, so that "Washington" will find the capital city of the US and not the state. To find the state, use "WA."

### Accuracy

The accuracy of the Manifold geocoder depends almost entirely upon the accuracy of the US streets geocoding database it uses. The Manifold Geocoding Database uses address data extracted from the US Bureau of the Census TIGER/Line data set. Although TIGER/Line is the federal standard for address accuracy used to support the Constitutional requirement of a census that counts every citizen, even this mammoth data set does not accurately capture all possible addresses in the US. Although TIGER/Line is updated every few years using a network of many Census Bureau field offices it does not capture address exceptions nor does it provide a satisfying level of geospatial accuracy in rural areas.

A further limitation of TIGER/Line is that due to constant churning of zip codes by the US Postal Service it is possible (although very rare) that a zip code for a valid geographic location might not be matched. A more
frequent problem is the appearance of zip codes within address records that are abstract zip codes (such as those assigned to some ships in the US Navy) that do not correspond to a geographical location within the United States.

If a street address cannot be found within the Manifold Geocoding Database, resulting in an unmatched record with no options presented, the user has several choices to deal with the unmatched address:

- Accept a zip code only match, if the zip code exists.
- Ignore the address.
- Manually create a point for the address in a drawing, if local knowledge is available.
- Use a geocoding system from another vendor to locate the address, and manually add latitude and longitude values to the geocoded table for that address record.

When using the geocoder for demographic or marketing studies it is usually safe to ignore unmatched addresses because addresses that can not be found in TIGER/Line are usually randomly dispersed. After all, if within a sample of 10,000 addresses one can obtain 90% geocoding without any effort to identify unmatched records the 9,000 data points thus obtained will normally be highly representative of the characteristics of the overall data set. For some applications, of course, achieving a match for every record may be a sufficiently important objective to merit a significant amount of time working with the Unmatched Records dialog or manual geocoding or use of a different geocoder.

Troubleshooting

If the geocoder does not work, check the following:

- Either the Manifold Geocoding Database or MapPoint or some other geocoding data source has been installed.
- All state files required are present (that is, required files have not been removed after installation).
- The Geocoding Database folder in the Tools - Options - File Locations correctly specifies the folder used to install the geocoding database. By default, this location is C:\Program Files\Manifold System\GCDB.
- If Manifold has been installed in a different folder than the default and if geocoding functions do not work within IMS, check that the geocoding database has been installed within the main Manifold installation folder, such as in the default path above.
- Are all columns used for addresses text columns?
- If MapPoint has also been installed, make sure the Use MapPoint … option has been checked.

If an address cannot be located, check the following:

- The address is correct. Nowhere is the ancient maxim of computing, "Garbage in, garbage out" more correct than in street address geocoding.
- The address does not contain secondary address information, such as "Apartment 20" co-mingled with the primary street address field.
- The address is a real street address and is not a virtual address, such as a Post Office box address (which, of course, can only be geocoded to a zip code).
- Try manually breaking up a single line address into street, city, state and zip fields.
- If the zip code cannot be found, check the zip code against the US Postal Service's website to make sure it is a valid zip code.
- If MapPoint has also been installed, make sure the Use MapPoint as… option has been checked in the Tools - Options - Geocoding dialog.

See Also

About Geocoding
Geocoding Data Sources
Geocoding with MapPoint
Geocoding Extensions
Geocoding Data Extensions
Geocoding Data Sources

The Geocoding Tools package is an optional extension to Manifold System that provides street address geocoding capability as well as access to the Manifold Geocoding Database data provided on the Manifold downloads site for both US street address geocoding as well as the easy creation of drawings showing US streets. The Geocoding Tools package also requires access to one or more geocoding data sources that are appropriate for the geographic region of interest.

Very important: the Geocoding Tools package will not be operational until you install at least one geocoding data source. You must install at least one geocoding data source.

A geocoding data source is a data source that provides the internal Manifold geocoding engine information about the location of street addresses. The geocoding engine can consult the data source to get information that helps the geocoding engine estimate the location of a particular street address. Once enabled by the installation of the Geocoding Tools package, Manifold can utilize the following geocoding data sources:

- The Manifold Geocoding Database provided for free download on the manifold.net site. It provides data for streets in the United States extracted from the US Bureau of the Census TIGER database. The Manifold Geocoding Database is fast, but is not as complete as the internal databases within Microsoft's MapPoint product. It is limited to the US only. The Manifold Geocoding Database is provided at no charge on the manifold.net site, so it is usually the first geocoding data source employed by new users in the US and more often than not the only geocoding data source users in the US ever need employ.
- Microsoft's MapPoint product in North American or European editions, which provide data either for North America or for selected countries in Europe. When MapPoint is installed on the same computer as Manifold System, the Manifold geocoder can reach into MapPoint's internal database to use it as a geocoding data source. MapPoint has one of the very best geocoding databases in the GIS industry, but it is significantly slower than using the Manifold Geocoding Database. Although MapPoint must be purchased, the combination of Manifold together with MapPoint is by far the least expensive way of doing street address geocoding for Canada and Mexico or for the countries supported by the MapPoint European edition.
- User-provided geocoding data extensions in either address range or points of interest formats. User-provided data sources must be used outside of the US or of the coverage provided by MapPoint. Even within the US or Europe, user-provided data sources are often used when exact precision is required since only "points of interest" formats can provide exact address precision.
- Web-based Manifold geocoding servers that provide geocoding services. Any such data source must fulfill the requirements of the Manifold Geocoding Server interface, but web services vary widely in their speed, accuracy and cost. Web services do have the potential advantage of greater timeliness, as it is possible to update a centralized data source more rapidly than, say, MapPoint editions may be updated. Visit the manifold.net web site for information on free geocoding server modules that may be downloaded.

Any combination of the above geocoding data sources may be used. This topic briefly surveys the above geocoding data sources and discusses the Manifold Geocoding Database in greater detail. See the specific Help topics for each of the other geocoding data sources for greater details.

A Geocoding Data Source Strategy

Users operating in the United States will normally use the Manifold geocoding database by default, since it is free. Users desiring slightly greater "hit" rates will also install MapPoint so MapPoint can be consulted if an address cannot be found in the Manifold geocoding database (the default setting).

Users in Canada or in the countries supported by MapPoint European edition don't really have much choice but to use MapPoint as a geocoding data source as this is generally far less expensive and easier to do than to try to acquire geocoding databases from other vendors and to convert such databases into the format required for user-provided geocoding data extensions.

Users outside the above areas cannot engage in street address geocoding unless they provide their own geocoding data extensions. That can be a difficult and expensive proposition. However, using Manifold on a notebook computer together with a portable GPS receiver, two employees in a vehicle (one to steer the vehicle and one to operate the notebook computer) can acquire addresses in points of interest format very rapidly. Many hundreds of streets can be done in a week and thousands of streets in well under a month by two people in a single vehicle. Once acquired, such data can be very valuable, the value of the data far exceeding the cost of acquiring it.
Web-based geocoding servers are so varied as to defy characterization. They range from very slow to very fast, from free to very costly, from very timely to remarkably obsolete. The Manifold Geocoding Server interface allows users to write modules that enable use of a very wide range of web-based geocoding servers.

**Precision**

Most users in the US doing street address geocoding with Manifold will use the Manifold US streets geocoding database provided as a free download on the manifold.net site. This database is based on the US government's official address database published in the TIGER/Line data set and allows the Manifold geocoder to find estimated positions for street addresses in the United States. It works best in urban and suburban areas where street addresses follow reasonably regular patterns.

**Very Important:** The Manifold Geocoding Database does not capture all possible street addresses nor can it guarantee the exact location of any address. Therefore, it must **not** be used for applications, such as 911 or other emergency response applications, which require every address to be exactly located. Note that the Manifold System End User License Agreement (EULA) specifically excludes uses, such as emergency response, that require fail-safe performance.

Nonetheless, the Manifold geocoder working together with the Manifold Geocoding Database is a good choice for many GIS tasks, such as demographic or marketing studies, that require geocoding of a reasonably high percentage of addresses. For such uses the Manifold Geocoding Database is highly effective and a spectacularly good value.

If Microsoft MapPoint is installed, the MapPoint database provides slightly greater accuracy in the United States, finding a few percentage points more addresses in large geocoding projects than is possible using the Manifold Geocoding Database. The MapPoint geocoder uses a variety of data sources in addition to government data to provide slightly better address recognition than is possible with government data only.

Some users prefer to use a Geocoding Data Server like the open source geocoding server module for Microsoft's Virtual Earth geocoding server. That has the advantage of frequently updated data in Virtual Earth, but it is slower since it has to go through Internet to geocode each address.

User-provided geocoding data extensions are normally used to provide custom information on street addresses not found within the Manifold Geocoding Database or MapPoint. For example, users may have specific address information licensed from commercial sources that provide more recent data than the TIGER data used in Manifold's geocoding database product or user may have custom address information for countries not covered by either Manifold or MapPoint.

**Installing the Manifold Geocoding Database**

Street address geocoding functionality within Manifold requires installation of an appropriate data source. If US street address geocoding is intended, we can use the Manifold Geocoding Database provided as a free download on the manifold.net site as our data source. If the database is not installed on the computer system and no other geocoding data source is provided, Manifold's street address geocoding commands and functions will not be available. The full US streets Manifold geocoding database requires approximately 950 MB of free space on disk.

**Installing the Manifold US geocoding database:**

1. Download the Manifold Geocoding Database zip file from the product downloads page in the manifold.net website. This is a **very large** file, over 800 MB in size. If your Internet connection is not totally reliable it might get damaged in transit. If it does not unzip correctly, download it again until it does. Unzip the zip file.
2. In Windows Explorer, drill down into the unzipped files to find the GCDB.msi installation program and double click on the GCDB.msi installation program to launch it. **Note:** If Windows Explorer has been set to hide extensions for well-known file types, this file may be listed in Windows Explorer as GCDB without the .msi extension. **Warning:** The geocoding installation .msi file is a huge file. Microsoft Windows Installer wants to check each .msi file before installing it so there may be a **very long** period of apparent inactivity while the Installer checks the GCDB.msi file, perhaps even tens of minutes on a slow machine. This is normal: just let the installation process continue to run and all will be well.
3. The installer will offer to install the geocoding database into a default folder, C:\Program Files\Manifold System\GCDB, within the default installation folder for Manifold System. If you have installed Manifold System in a different location on your hard disk it is strongly recommended that you update the file locations for geocoding in any IMS config.txt file so that geocoding capabilities will be available within Manifold IMS.
4. Launch Manifold. In the **Tools - Options - File Locations** pane, specify the folder used to install the geocoding database for the **Geocoding Database** folder and press **OK**. This step may be skipped if the default location for installation is used.

The **states.dat** file must always be available on your hard disk. The geocoding database is organized by US states with a file for each state. Each state file ends in a .dat extension and is named using the state postal abbreviation. For example, the geocoding database files for California and New York are **ca.dat** and **ny.dat** respectively.

The **GCDB.msi** installer will install geocoding database files for all US states. However, only those state files for which addresses will be geocoded need be located on the hard disk. Files for states that will not be used may be removed to free up disk space. For example, if we will be geocoding street addresses only in the state of California and no other state, we may delete all files from the **GCDB** folder on hard disk except the **states.dat** file and the **ca.dat** file.

**Tech Tip:** Don’t delete any of the files. In an era of hard disks of virtually limitless size, there’s little point in trying to save a few megabytes by deleting state files and at the same time much risk of inconvenience should those state files be needed at some future date.

To uninstall the US geocoding database and entirely remove all files, we may use the Windows Control Panel **Add / Remove Programs** applet. Removing the geocoding database will render all geocoding functions inoperative.

**Geocoding Data Extensions**

User extensions to street address geocoding data may be provided in the form of a Manifold .map project file that contain tables with columns using predefined names and types. The project file must be placed in the folder cited in the **Tools - Options - File Locations** pane for **Geocoding Extensions** and the **Use geocoding extensions** checkbox must be checked in the **Tools - Options - Geocoding** pane.

User geocoding extensions are always considered first when processing data. Next, either the Manifold Geocoding Database or MapPoint geocoding data are consider depending on whether MapPoint is set to be used before or after the Manifold geocoding database as specified in the **Tools - Options - Geocoding** dialog.

See the Geocoding Data Extensions topic for details on user-supplied extensions.

**See Also**

- About Geocoding
- Geocoding Tools
- Street Address Geocoding
- Geocoding with MapPoint
- Geocoding Data Extensions
- Manifold Geocoding Servers
Geocoding with MapPoint

Microsoft's MapPoint product is a high quality "atlas" mapping program that is popular for many consumer applications. MapPoint also has the ability to geocode street addresses within the region covered by the edition of MapPoint in use. The European edition of MapPoint can geocode street addresses in eleven European countries (Austria, Belgium, Denmark, France, Germany, Italy, Luxembourg, the Netherlands, Spain, Switzerland and United Kingdom).

If the Manifold Geocoding Tools package is installed, Manifold System can use MapPoint as a geocoding data source to enable the Manifold geocoding engine to provide street address geocoding capabilities. See the Geocoding Tools topic for information on the capabilities provided by the Geocoding Tools package. If the Geocoding Tools package is not installed, the capabilities provided by this topic will not be available.

If a recent edition of Microsoft MapPoint (from 2002 onward) is installed on the same computer as Manifold System, Manifold can use the database within MapPoint to provide a geocoding data source for Manifold. MapPoint can be used as an adjunct to the Manifold Geocoding Database or it can be used as a replacement for the Manifold Geocoding Database. See the Geocoding Data Sources topic.

Most geocoding functions, including geocoding extensions in SQL, programming and use within IMS will work (IMS usage will require correct configuration of user permissions - see below). However, note that functions with MapPoint will be limited to those that use a full street address.

Although the Manifold Geocoding Database enables geocoding down to the centroid of the lowest resolvable entity (state, city, zip code) if a full street address is not available, MapPoint cannot geocode to, say, the centroid of a postal code if no more detailed street address is known. In general, MapPoint will not return the centroid of a postal code. It will return the location of a valid street address, and will "jump" to the location of a postal code in the MapPoint user interface when used interactively. However, MapPoint will not return the location of a postal code when it is called programmatically as is the case when it is used as a data source for Manifold.

Because MapPoint is much less expensive than buying proprietary geocoding data sets for Canada or European countries, using MapPoint as a geocoding data source is the least expensive way of getting Canadian or European street address geocoding within a true GIS like Manifold. Since both the North American and European editions of MapPoint may be simultaneously installed, Manifold users who have both editions installed can simultaneously geocode street addresses in the US, Canada and in the European countries supported by MapPoint.

MapPoint may also be used as the default geocoder for US street addresses and MapPoint may be used as a geocoding data source for Manifold even if the Manifold Geocoding Database is not installed. Although MapPoint does cost more than the Manifold Geocoding Database, which is provided for free download on the manifold.net web site, it provides more extensive and more up-to-date street address matching capability for US street addresses. The downside of using MapPoint is that geocoding will run slower because accessing information within MapPoint as a geocoding data source is slower than using the Manifold Geocoding Database.

Installation and Use

Using MapPoint as a geocoder is easy: simply install the edition(s) of MapPoint desired, and then in the Tools - Options - Geocoding dialog check the Use MapPoint ... option and in the same dialog choose whether MapPoint is to have priority over the Manifold geocoding database. Most users with large geocoding tasks in the US will give the Manifold geocoding data priority (the default setting) so that geocoding can run as fast as possible with only those addresses not found in the Manifold geocoding database subjected to the slower MapPoint data lookup.

With MapPoint installed and used as the geocoder, the geocoding process works just as it would when using a Manifold street address data product or when using your own geocoding data extensions. Standardization works the same as well, with any auxiliary information returned by MapPoint in the process of standardizing an address being appended to the appropriate column.

When using MapPoint for geocoding, country names are specified as follows:

- If a single field is used for addresses the various parts of the address (street address, city, etc.) must be separated by commas. The MapPoint address parser in general cannot extract address parts (street address, city, state, zip/postal code or country) from a single field unless they are separated by commas.
- If separate fields are used, then the **State** field must contain the state/province name and the **Country** field should contain the country name.
To geocode Canadian or European Addresses using MapPoint

1. Install the required edition(s) of MapPoint.
2. Launch Manifold and in Tools - Options - Miscellaneous check the Use MapPoint ... option. Check the Give MapPoint priority... box if MapPoint is to be the primary (or only) geocoder and uncheck that box if the Manifold geocoder is to be used as the primary geocoder.
3. If geocoding addresses outside of the United States, include the country name in a Country field. Alternately, if only one country will be used, specify the default country name in the Tools - Options - Geocoding dialog.
4. Follow the instructions in the Street Address Geocoding topic. All Manifold geocoding functions will operate correctly using MapPoint as the geocoder.

Examples

Suppose we would like to geocode the street addresses of some hotels in Vancouver, Canada.

### Single Field Addresses

<table>
<thead>
<tr>
<th>Addresses</th>
</tr>
</thead>
<tbody>
<tr>
<td>645 Howe St, Vancouver, BC, Y6C2Y9, Canada</td>
</tr>
<tr>
<td>1128 West Hastings Street, Vancouver, BC, V6B4R5, Canada</td>
</tr>
<tr>
<td>1277 Robson St, Vancouver, BC, V6E1C4, Canada</td>
</tr>
<tr>
<td>845 Burrard St, Vancouver, BC, V6Z2K6, Canada</td>
</tr>
<tr>
<td>567 Hornby St, Vancouver, BC, V6C2E8, Canada</td>
</tr>
<tr>
<td>322 Davie St, Vancouver, BC, Y6B5Z6, Canada</td>
</tr>
</tbody>
</table>

If our table contains the addresses in a single field, we must take care to separate the different parts of the address using commas. Before a single-field address can be used with MapPoint it must be split into multiple fields using Standardize.

### Multiple Field Addresses

<table>
<thead>
<tr>
<th>Address</th>
<th>City</th>
<th>State</th>
<th>Zip</th>
</tr>
</thead>
<tbody>
<tr>
<td>645 Howe St</td>
<td>Vancouver</td>
<td>BC, Canada</td>
<td>Y6C2Y9</td>
</tr>
<tr>
<td>1128 West Hastings Street</td>
<td>Vancouver</td>
<td>BC, Canada</td>
<td>V6B4R5</td>
</tr>
<tr>
<td>1277 Robson St</td>
<td>Vancouver</td>
<td>BC, Canada</td>
<td>V6E1C4</td>
</tr>
<tr>
<td>845 Burrard St</td>
<td>Vancouver</td>
<td>BC, Canada</td>
<td>V6Z2K6</td>
</tr>
<tr>
<td>567 Hornby St</td>
<td>Vancouver</td>
<td>BC, Canada</td>
<td>V6C2E8</td>
</tr>
<tr>
<td>322 Davie St</td>
<td>Vancouver</td>
<td>BC, Canada</td>
<td>Y6B5Z6</td>
</tr>
</tbody>
</table>

If our table contains the addresses as multiple fields, we must make sure that the field to be used as the State field includes the province and country separated with a comma. Standardize will automatically split single field addresses into multiple field addresses that correctly have the province (if used) and country in the State field.
Running the geocoder will add **Latitude** and **Longitude** columns and values to the table.

### Multiple Field Addresses

<table>
<thead>
<tr>
<th>Address</th>
<th>City</th>
<th>State</th>
<th>Zip</th>
</tr>
</thead>
<tbody>
<tr>
<td>645 Howe St</td>
<td>Vancouver</td>
<td>Canada</td>
<td>V6C2Y9</td>
</tr>
<tr>
<td>1228 West Hastings Street</td>
<td>Vancouver</td>
<td>Canada</td>
<td>V6B4R5</td>
</tr>
<tr>
<td>1277 Robson St</td>
<td>Vancouver</td>
<td>Canada</td>
<td>V6E1C4</td>
</tr>
<tr>
<td>845 Burrard St</td>
<td>Vancouver</td>
<td>Canada</td>
<td>V6Z2K6</td>
</tr>
<tr>
<td>567 Hornby St</td>
<td>Vancouver</td>
<td>Canada</td>
<td>V6C2E8</td>
</tr>
<tr>
<td>322 Davie St</td>
<td>Vancouver</td>
<td>Canada</td>
<td>V6E5Z6</td>
</tr>
</tbody>
</table>

Note that MapPoint can almost always find a street address even if the province is not included in the **State** field. However, the country name must always be included in the **State** field so MapPoint knows which country to use.

### Postal or ZIP Codes

The **Standardize** command will extract address, city, state and postal/ZIP fields from an address given in a single field. It also works when MapPoint is used as a geocoder except that MapPoint will not extract and return the postal or ZIP code from a single address field. Therefore, when converting a single-field address to a multiple field address using **Standardize** the postal/ZIP code field will be blank.

For this reason, if postal codes are required to geocode street addresses in a given area and MapPoint is the geocoder, use addresses in multiple field form. Note that usually having a street address and a city is usually enough for geocoding purposes since in most cases the postal/ZIP code is redundant if the city name and name of the street are known. In cases where the postal/ZIP code is required and addresses are in single field form, it us usually easy to use the transform toolbar token operators to manually extract postal/Zip codes from a single field address into a separate Zip field.

When using geocoding functions in SQL, the **LocateZipLat** and **LocateZipLon** functions only work for US zip codes, something to keep in mind when using MapPoint European edition in Europe.

### MapPoint Versions

The Microsoft product meant by "MapPoint" in this topic is the packaged consumer application that installs on the local computer, and not the .NET service offered by Microsoft. The MapPoint geocoding facilities used by Manifold System require a reasonably current edition of MapPoint from 2002 onward and will not work with earlier MapPoint editions.

The European edition of MapPoint 2004 has enhanced support for additional European countries. Following is a list of countries covered by European editions of MapPoint 2004 and MapPoint 2002 for street address geocoding:

- Austria (partially covered by both 2004 and 2002).
- Belgium (fully covered by 2004, partially covered by 2002).
- Denmark (fully covered by 2004, partially covered by 2002).
- Finland (partially covered by 2004, not covered by 2002).
- France (partially covered by both 2004 and 2002).
- Germany (fully covered by both 2004 and 2002).
- Italy (partially covered by both 2004 and 2002).
- Norway (partially covered by 2004, not covered by 2002).
- Portugal (partially covered by 2004, not covered by 2002).
- Spain (partially covered by both 2004 and 2002).
- Sweden (fully covered by 2004, not covered by 2002).
- Switzerland (fully covered by 2004, partially covered by 2002).
- United Kingdom (partially covered by both 2004 and 2002).

MapPoint 2004 also covers Czech Republic and Ireland, but only in that those countries can be used during routing. They cannot be used for street address geocoding.

Expectations are that MapPoint editions will continue to be compatible with the interface used by Manifold to access MapPoint as a geocoding data source.

**MapPoint and IMS**

MapPoint geocoding will normally not work from IMS unless we map anonymous Internet connections to a user account that has more enhanced permissions than the default Internet access `IUSR_xxx` account.

**To enhance permissions for the Internet access account:**

1. Create a regular user account with default permissions that is a member of the `Users` group.
2. Open Control Panel - Administrative Tools - Internet Information Services (or the equivalent dialog in your version of Windows).
3. Right click the folder that contains the published website, select Properties, switch to Directory Security and click Edit under Anonymous access and authentication control.
4. Set the user account used for anonymous access to the account created in step 1 above.

**See Also**

About Geocoding
Geocoding
Street Address Geocoding
Geocoding Data Sources
Geocoding Data Extensions
Geocoding Data Extensions

If the Manifold Geocoding Tools package is installed, Manifold System can use various geocoding data sources to enable the Manifold geocoding engine to provide street address geocoding capabilities. See the Geocoding Tools topic for information on the capabilities provided by the Geocoding Tools package. If the Geocoding Tools package is not installed, the capabilities provided by this topic will not be available.

Users may customize geocoding operation by providing their own data to be used as a geocoding data source. Such additional, customized data are referred to as geocoding data extensions because most frequently such user-provided data is intended as an adjunct to existing geocoding data sources, perhaps providing corrections or a limited amount of more recent data. However, even though such data is referred to as extensions, Manifold can use user-provided data without any other geocoding data source required.

Users provide geocoding data extensions in the form of one or more Manifold .map project files that contain tables with columns using predefined names and types. The project files must be placed in the folder cited in the Tools - Options - File Locations pane for Geocoding Extensions and the Use geocoding extensions checkbox must be checked in the Tools - Options - Geocoding pane.

User geocoding extensions are always considered first when processing data. Next, either the Manifold geocoding database or MapPoint geocoding data are considered depending on whether MapPoint is set to be the primary or the fallback geocoding engine.

Geocoding data extensions are used mainly in two forms:

- First, they may be used to provide addresses for specific points of interest that exist at a specific latitude or longitude (such as the address and location of a hotel or restaurant). This format is easier to understand and implement.
- Second, they may be used to provide ranges of addresses such as the ranges of address that may occur on a given street segment so that any address and location within the range may be interpolated.

The format for data extensions allows for both types of uses.

Tables used for geocoding data extensions may be local within the project, they may be linked from external DBMS providers or, if Enterprise Edition is in use, they may be shared from an Enterprise server.

Format

Each .map file must have a table named Geocoding Data. The table must have one or more address columns using specific names (Address, City, Country, From Number, Number, State, Street, To Number, Zip) and location columns that give latitude and longitude locations. The location columns may be either latitude and longitude columns (Latitude, Longitude) that specify a single location or they may be range columns (From Latitude, From Longitude, To Latitude, To Longitude) that specify a range of locations.

Address columns except Number columns (Number columns include Number, From Number and To Number) must be text. Number columns may be text or numeric, and latitude and longitude columns must be numeric. Latitude and longitude coordinates are assumed to be in WGS 84.

Range latitude and longitude columns (From Latitude, From Longitude, To Latitude, To Longitude) are always used in conjunction with range Number columns (From Number, To Number) to geocode street segments. The Street and Number columns are used as a substitute for the Address column if there is no Address column. If there is an Address column, the Street and Number columns are ignored.

Examples

Suppose we would like to create a user-supplied geocoding data extension that contained points of interest. For each point of interest such as a hotel or restaurant we have a record that contains a specific address and latitude / longitude location.

We would create a table called Geocoding Data and make sure that it contains the following fields and field types:
- State - Text (ANSI, variable-length)
- City - Text (ANSI, variable-length)
- Address - Text (ANSI, variable-length)
- Latitude - Latitude
- Longitude - Longitude

In the above table each record represents a specific address at a specific location. When a given address is geocoded the exact latitude and longitude will be reported. If an address can not found in the table then no geocoded latitude and longitude will be reported.

Suppose we would like to create a geocoding data extension table for ranges of addresses. We could create a table called **Geocoding Data** and make sure that it contains the following fields and field types:

- State - Text (ANSI, variable-length)
- City - Text (ANSI, variable-length)
- Street - Text (ANSI, variable-length)
- From Number - Integer (32-bit)
- To Number - Integer (32-bit)
- From Latitude - Latitude
- From Longitude - Longitude
- To Latitude - Latitude
- To Longitude - Longitude

In the above table each record represents a street segment that extends from a **From** latitude and longitude location to a **To** latitude and longitude location. The street segment contains addresses numbered from the **From** number to a **To** number. For example, a street segment may contain addresses from 100 to 200 on a **Street** called Western Avenue. An address located at 150 Western Avenue will be located at the middle of the straight line drawn from the **From** latitude and longitude location to the **To** latitude and longitude location.

The **From Number** used in a street segment may be greater than the **To Number**. More than one record can overlap for the same **Street** name with **From Number** and **To Number** address range so long as the overlapping number ranges are all odd or all even. The geocoding engine can exploit an all odd or all even number range to assign an odd or even street address number to the correct segment.

Consider a situation where a street segment has a **From Number** of 10 and a **To Number** of 31. This is a typical situation where a building number between 10 and 31 with that same street name will be geocoded within that segment.
If desired, the same street can be represented by more than one street segment where the ranges overlap numerically but the segments are distinguished by having all odd or all even number ranges. In the above illustration the same portion of the street is represented using three segments. There are two segments that use all even numbers for the From Number and To Number values (10 to 20 on one segment and 24 to 30 on the other segment). There is one segment that uses all odd numbers (11 and 31) for the From Number and To Number values.

The illustrations above show the position of street segments as lines for conceptual clarity, but .map files used for geocoding data extensions need not contain any drawings. All they need contain are one or more tables that have either point locations (as in the case of points of interest) or ranges of locations.

When matching a building number (street address number) against street segments for that street name, if the building number is even the geocoding engine will first try to find a segment into which the number falls where the number ranges are both even. If the building number is odd the engine will try to find a segment into which the number falls where the number ranges are both odd. For example, given three segments for Main Street with From / To Number ranges of 10 to 20, 24 to 30, and 11 to 31, the geocoding engine will match 18 Main Street to the first segment and 25 Main Street to the third segment.

If the geocoding engine cannot find any all odd or all even segments into which the street number falls, it will try to find a segment with a mixed odd/even From / To range (like the first illustration above) into which the number falls.

Note that the above table organization makes two assumptions:

- There is no distinction between the left or right sides of the street in address ranges. Therefore, left and right offsets should not be used when geocoding using user-supplied geocoding data extensions. In some cases, left and right side addressing can be accomplished if the left or right sides of the street are exclusively odd or even numbers. In such cases, use two segments for the same part of the street with one segment being odd numbers and the other segment being even numbers. These segments may be offset from the centerline, as in the second illustration above, to create the effect of a left or right offset.
- Each street segment geocoded is a straight line. A highly curved street should therefore be segmented into straight line segments that approximate the curves of the street with address ranges and latitude / longitude location ranges assigned for each segment.

In many cases, when creating custom geocoding data it may be easier to simply use points of interest organization and assign a specific location to each address that actually exists in the area of interest. Using the GPS Console or other GPS software it is remarkably fast and easy to collect street address locations even for very large numbers of addresses. Points of interest organization is also usually much more accurate than street segment ranges when creating custom geocoding data extensions for rural areas.

**Tech Tip**

Keep in mind that Manifold may lock the .map files used for geocoding data extensions or may cache their content; therefore, any modifications made to such files will only be visible in sessions of Manifold launched after any such modifications.

**See Also**
About Geocoding
Geocoding
Geocoding Tools
Street Address Geocoding
Geocoding Data Sources
Geocoding with MapPoint
Manifold Geocoding Servers
Manifold Geocoding Servers

The Manifold street address geocoding engine can geocode addresses using generic geocoding data sources. A generic geocoding data source is a .NET object that supports the Manifold Geocoding Server (MGS) interface. When such a .NET object is correctly written and made available within a Manifold installation, it is called a Manifold Geocoding Server, that is, it is a geocoding server that conforms to the interface which allows it to be used by Manifold System.

Although geocoding servers are often web-based geocoding services, there is no reason why the MGS interface could not be used to provide access by Manifold to a local application, such as some third-party commercial geocoding software. Part of the magic of .NET and the MGS interface is that they can be used to connect different applications which otherwise could not interoperate.

Required Geocoding Tools

Manifold's street address geocoding capability becomes operational only when the optional Manifold Geocoding Tools package is installed. If you do not have the optional Geocoding Tools package, you will not be able to use the geocoding servers capability discussed in this topic.

Note that Manifold Universal Edition automatically installs the Geocoding Tools package. If you are using Universal Edition, you do not need to perform an extra Geocoding Tools installation as it has already been installed.

Enabling a Manifold Geocoding Server

When a .NET module for a Manifold Geocoding Server has been correctly written and made available to Manifold, it will appear in the Tools - Options Geocoding page. The pane captioned Use the following Manifold Geocoding Servers will show a list of available geocoding servers with checkboxes to turn them on or off. Check the box of a geocoding server to turn it on.

By default, Manifold will have one Manifold Geocoding Servers listed, for Oracle geocoding with Oracle Spatial. This geocoding server requires installation of Oracle Spatial geocoding capability. It is provided as a "built-in" capability of Manifold to assure compatibility with Oracle Spatial DBMS products.

Earlier editions of Manifold also had two additional Manifold Geocoding Servers listed but these have since been moved out into optional downloads:

- The free Geocoder.us web-based service.
- Microsoft Virtual Earth geocoding using the Virtual Earth service.

Because web-based geocoding servers often change, a geocoding server built into Manifold cannot be maintained without a new release of Manifold. Therefore, the geocoding servers for Geocoder.us and for Virtual Earth have been moved into free modules that can be downloaded from the manifold.net web site. Drill down through the Product Downloads page to find these.

The .NET module for the Oracle geocoding servers is automatically installed when Manifold System is installed. To get the geocoding servers for Geocoder.us and Microsoft Virtual Earth, download the zip file containing them from the manifold.net web site and unzip them to get the .dll files that make up the geocoding server modules. A geocoding server module is simply a .dll file. Place the .dll files for the geocoding servers into your Manifold System installation folder. When you restart Manifold they will be automatically incorporated into Manifold.

The default Manifold System installation folder is C:\Program Files\Manifold System. If you are operating a 64-bit Manifold installation in 64-bit Windows or a 32-bit Manifold installation in a 64-bit Windows system there will also be a second Manifold System installation folder at C:\Program Files (x86)\Manifold System. In that case, the .dll files for image server modules should be placed in both the C:\Program Files\Manifold System and also in the C:\Program Files (x86)\Manifold System installation folders.

When Manifold launches, none of the installed geocoding server modules are checked by default: if we would like to use a particular geocoding server we must launch the Tools - Options dialog, click on the Geocoding page and then check the box for the desired geocoding server.

Example: Using the Geocoder.us Geocoding Server
Consider the table of sushi restaurants from the About Geocoding topic.

<table>
<thead>
<tr>
<th>Sushi Restaurants *</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Sushi Ya</td>
</tr>
<tr>
<td>Miyaki</td>
</tr>
<tr>
<td>Higashi West</td>
</tr>
<tr>
<td>Naoko Sushi</td>
</tr>
<tr>
<td>Tengu Sushi</td>
</tr>
<tr>
<td>Minokichi</td>
</tr>
<tr>
<td>Le Poisson Japonais</td>
</tr>
<tr>
<td>Jidai-Ya</td>
</tr>
<tr>
<td>Akasaka</td>
</tr>
<tr>
<td>Mikasa</td>
</tr>
<tr>
<td>Toshi's Sushiya</td>
</tr>
</tbody>
</table>

We can geocode this table using the Geocoder.us geocoding server as follows:

In the Tools - Options dialog's Geocoding page we check the box for the Geocoder.us geocoding server and we uncheck all other boxes, such as the Use Manifold Geocoding Database box, for other geocoding data sources. We do this for the sake of this example to guarantee that any geocoding will be done only via the Geocoder.us geocoding server.

Next, with the table open we launch the Table - Address - Geocode dialog and geocode the table as we normally would. The process is remarkably slow, but eventually it does indeed geocode all of the records.

Advantages and Disadvantages of Geocoding Servers

This simple example shows some of the advantages and disadvantages of using a geocoding server. On the plus side, this particular geocoding server is free. Because it is based upon a centralized web database, it has the potential to use a geocoding database that is timelier than a geocoding database that is resident on your hard disk. It is possible to update a database on a web server continuously, while downloading files can take time. Therefore, a web-based geocoding service has the potential to incorporate changes in street addresses, such as the inclusion of new streets in rapidly growing regions, more rapidly than geocoding data sources distributed as files.

This advantage may at times result in a geocoding server returning a different location than a location determined using the Manifold Geocoding Database or other geocoding data sources. In the above example, the latitude and
longitude locations obtained using the Geocoder.us geocoding server were identical to those obtained using the Manifold Geocoding Database. That's not surprising, because both the Manifold Geocoding Database and Geocoder.us use the same underlying TIGER/Line 2004 data. However, it could well be that a geocoding server using a different database would yield different locations.

This example also raises some typical limitations of web-based geocoding services. First, although it is theoretically possible to keep the service’s database more up to date than a downloaded distribution of geocoding data, in actual practice few organizations have the resources to do more than simply recycle TIGER data. As of this writing, the free Geocoder.us web service uses the same TIGER/Line 2004 database used in the Manifold Geocoding Database so it will not be able to find any addresses not found in the Manifold Geocoding Database nor will it be able to plot any found addresses more accurately than the Manifold Geocoding Database.

Second, the free service is very slow. Using the Manifold Geocoding Database to geocode eleven records takes no time at all: the records are geocoded virtually as soon as we finish clicking the OK button in the Geocode dialog. In contrast, using the Geocoder.us server requires over two minutes for eleven records. Such a slow pace means that over four hours will be required to geocode a mere 1,000 records. As a practical matter, this speed limitation will prevent the use of the free Geocoder.us service in commercial applications, no doubt the intention, as a free demo service is not usually considered a substitute for faster, paid commercial services.

Third, the quality and performance of a web service can vary greatly between free versions, if such are offered, and paid versions of the same service. Some paid versions are very expensive compared to simply using the Manifold Geocoding Database or using MapPoint as a supplementary geocoding data source. Many commercial geocoding web services will offer a free version that is slower than the paid service or which is limited to some smaller number of records. Such limitations can cause frustration or time wasted on attempted troubleshooting if users are not aware of those limitations. If users understand the limitations of free services, such services can be an effective way of testing the range or accuracy of a given geocoding server against a sample set of data before purchasing a paid service.

Finally, geocoding web services can apply all sorts of restrictions to their use and not all sites do a good job of setting forth the terms and conditions of use. Some web sites attempt to specify limitations using language that is not drafted by legal professionals who are expert in such matters, or (at times a worse situation) using language drafted by legal professionals who may understand the law perfectly well but who are clueless about technology. Either way, the result can be language that is ambiguous, is nonsense or is self-contradictory within the context of the technology being used. That can force users into a guessing game as to what uses are allowed or not allowed. In all fairness, this is not a problem just with geocoding web services but seems to be a generic problem in many web sites.

The Geocoder.us web site as of the present writing appears to limit free access to "non-commercial" purposes. That is an easy restriction to observe for Manifold users since the free Geocoder.us service is too slow for commercial use, whatever is meant by that term. Even if it was faster, because the data behind Geocoder.us is the same as the data in the Manifold Geocoding Database (provided for free download on the manifold.net web site) and the Manifold Geocoding Database may be used for commercial purposes, there is no reason why anyone would choose to use Geocoder.us instead of the Manifold Geocoding Database. That is especially true given that using the Manifold Geocoding Database as a geocoding data source works immensely faster than using Geocoder.us.

Nonetheless, because Geocoder.us is a good idea and deserves publicity, it was selected as an example for implementation of a Manifold Geocoding Server. Users who experiment with the free Geocoder.us service will be more likely to sign up for paid service, which most likely will be faster. In addition, open source experimentation like that involved in people cobbling up their own examples of geocoding servers using the MGS interface is the sort of thing that helps new web sites like Geocoder.us get traction and expand their user base.

Continued user interest in this area will help web-based geocoding servers get better and better. We expect that in the future such servers will emerge to provide geocoding services, perhaps for a fee, in areas outside of the United States or to provide specialized geocoding services such as rapidly updated or especially accurate geocoding data sources using points of interest data. We hope the examples published by manifold.net will help such activities to flourish.

The Manifold Geocoding Server Interface

See the Geocoding Server Interface topic for a description of the interface. Programmers should also visit the manifold.net website for any supplemental documentation or example implementations, if any are made available in the future. Source code is normally provided for any free geocoding server modules available for download on the manifold.net web site.

Troubleshooting
In case of difficult using a geocoding server, the first things to check are:

- Has the Manifold Geocoding Tools package been installed? Manifold has no street address geocoding capability if the Geocoding Tools package has not been installed.
- Has the desired geocoding server checkbox been checked in the Tools - Options - Geocoding dialog? Geocoding servers are not enabled by default and must be enabled by checking their checkbox.
- If the geocoding server is a web service, does the computer have Internet access? If you can't launch a browser to visit that geocoding server's web site (they almost always have some web page advertising the service) then Manifold will not be able to connect over the web to the service, either.
- Does this geocoding server require special configuration? Some geocoding servers may utilize services that require specification of user logins, passwords, account numbers or other information within some configuration file or through other means.

Manifold Geocoding Servers are not enabled by default for several reasons. The most important is that they are usually web-based services and web connections may not be available in all circumstances on machines where Manifold is installed.

Since geocoding servers are usually services on remote machines and the availability or manner of operation of such services is beyond the control of the user, it may be necessary to check if a particular service is still available before enabling it in Manifold or to configure some accessory files, perhaps to specify some special login name and password for a paid service, before the geocoding server may be used. Contact your provider of the geocoding server module being used for any such configuration information.

In the particular case of the Geocoder.us service for which an MGS module is provided as an example, although the service was available for free and worked fine at the time of publication of this release of Manifold, there is no guarantee that it will continue to be available to users at no charge or in the same manner. Web site URLs change frequently, so users should be alert to the possibility that a geocoding web service may have changed and is no longer available as a particular geocoding server module.

If you encounter trouble while using a third party Manifold Geocoding Server module that is said to enable a particular geocoding service, consider the possibility that the web service has changed and that a modified MGS module might now be required. Contact the author of the module being used and see if an updated module has been made available.

See Also

About Geocoding
Geocoding
Geocoding Data Sources
Geocoding Data Extensions
Geocoding Server Interface
Geocoding Tools
Geocoding with MapPoint
Street Address Geocoding
Spatial Geocoding with Match

The Table - Match command provides spatial geocoding within Manifold System. See the Geocoding topic for a general discussion of geocoding and the difference between address geocoding and spatial geocoding.

Match can be used to geocode tables by matching locations to drawings of postal codes, provinces, regions, counties or other geographic entities. Such drawings/maps are much more frequently available in international settings than are detailed data sets necessary for address geocoding. In addition, Match can be used to spatially geocode tables of all types using drawings as a guide. For example a table of well drilling information can be geocoded using a drawing of well locations as a guide.

Controls

**Match** - Choose a drawing to which the table should be spatially matched.

**Move to Top** - Move the highlighted field to the top of the field list. This gives it the highest priority in matching.

**Move Up** - Move the highlighted field up one position in the field list.

**Move Down** - Move the highlighted field down one position in the field list.

**Move to Bottom** - Move the highlighted field to the bottom of the field list. This gives it the lowest priority in matching.

**Use Case** - Consider upper or lower case when matching text values in fields. When pressed, "nAme" will not match "Name" or "name".

**Use Side Whitespace** - Consider space characters occurring before or after text strings. If pressed, " 89701" will not match " 89701" (with two spaces before the 8).

**Use Interior Whitespace** - Consider space characters occurring within text strings. If pressed, "8 9 7 0 1" will not match "89 701".

**Match All** - If pressed, requires a match to all fields.

**Column / Match To Pane** - Choose fields that must be matched to guide a spatial match.

**X** - Field name to save longitude / x data. Suggested name: "Longitude".

**Y** - Field name to save latitude / y data. Suggested name: "Latitude".

**Latitude / Longitude coordinates** - Write location in decimal degrees latitude and longitude.

**Match** is easy to use. If we have a database table that has one field in common with a drawing, we can use the drawing to geocode the table.

To spatially geocode a table using Match:

1. Open the table.
2. Choose Table - Match - Drawing
3. In the Match Drawing dialog choose the name of the drawing to serve as a spatial guide.
4. Choose the fields in the table that are to be matched to fields in the drawing.
5. Check boxes as desired for matching text values.
6. In the X and Y boxes specify a name to use for the longitude field and latitude field.
7. Verify the Latitude / Longitude coordinates box is checked and press OK.

Each record in the table will be compared to all objects in the drawing for the specified fields. If the record matches field values for an object in the drawing the latitude and longitude of that object's centroid will be written into the record's latitude and longitude fields.

If only one field is specified, the records in the table will be geocoded by matching objects against that single field. If two or more fields are specified, Manifold will try to match against all fields.

Multiple fields are used for sub-matches to resolve duplicates. For example, if we are geocoding a table that has a city name and a country name for each record there may be multiple cities of the same name in different countries (for example, a "Paris" in France as well as a "Paris" in the United States). If both the city name and the country name are specified (with the city name field listed above the country name field in the Match dialog), then Match will first match all city names and then match by country names to resolve city name duplicates.

Example

We have a ZIPs drawing that shows zip code areas using Census Bureau ZCTAs. We also have a Customers table that has a zip code field for each customer. The zip code field is a text column. We want to geocode the table so that each record acquires latitude and longitude values to position it at the zip code centroid for that customer.

The drawing used is called ZIPs. The ZIPs drawing shows Census Bureau Zip Code Tabulation Areas (ZCTAs) in the San Francisco Bay region. ZCTAs are standardized areas that approximate the coverage of particular zip codes. The drawing was created by selecting ZCTA's near the Bay from a drawing of all ZCTAs in California, copying them and then pasting them to their own drawing. The areas have been colored using the Color dialog for no reason other than to avoid a bland, default look.
If we click open the ZIPs drawing's table we see that for each area in the drawing we have one field, a ZIPCODE field that gives the ZIP code for each area. (For our international readers, "ZIP" codes are postal codes in the US.)

<table>
<thead>
<tr>
<th>Name</th>
<th>Zip</th>
</tr>
</thead>
<tbody>
<tr>
<td>Laurence Leibman</td>
<td>94060</td>
</tr>
<tr>
<td>Elizabeth Lincoln</td>
<td>94070</td>
</tr>
<tr>
<td>Victoria Ashworth</td>
<td>94063</td>
</tr>
<tr>
<td>Patrick Simpson</td>
<td>94062</td>
</tr>
<tr>
<td>Francisco Chang</td>
<td>94303</td>
</tr>
<tr>
<td>Yang Wang</td>
<td>94025</td>
</tr>
<tr>
<td>Pedro Afonso</td>
<td>94061</td>
</tr>
<tr>
<td>Elizabeth Brown</td>
<td>94027</td>
</tr>
</tbody>
</table>

If we click open our Customers table we see it has only two fields, a Name field giving the name of the customer and a Zip field giving the customer's zip code. Of course, a real "customers" table would likely have many more fields than just these two. This example shows only two fields to keep the illustrations as simple as possible.

To spatially geocode the table by matching it to the ZIPs drawing we choose Table - Match. This launches the Match Drawing dialog.

In the dialog we choose ZIPs as the drawing to match. To match using the Zip field we double-click into the Match To column for the Zip field and choose ZIPCODE in the list box that appears. Press ENTER to accept the ZIPCODE choice.
The Customers table does not yet have any fields in it that store the longitude and latitude value. We now tell Manifold what to name the fields that will be created. In the X box we enter Longitude and in the Y box we enter Latitude. Manifold will use these names for the X coordinate (the longitude) and for the Y coordinate (the latitude) values. We could name these fields whatever we choose but it is wise to name them "Longitude" and "Latitude" so that the table’s column names are self-documenting.

The ZIPs drawing is not a projected drawing so automatically the coordinates taken from it are degree-based latitude and longitude coordinates. If the drawing being used for the match was a projected drawing, the Latitude / Longitude coordinates checkbox would be enabled and would be checked by default.

Checking this box when using projected drawings will geocode the table records using degree-based latitude and longitude coordinates. Unchecking this box when matching to a projected drawing will geocode the table’s records using whatever native coordinate system is used in the projected map. For example, if one matches to a drawing projected into UTM coordinates and unchecks the Latitude / Longitude coordinates box the coordinates written into the table will be UTM coordinates. This is a rarely used option designed for expert usage.

The result seen in the table is that the Customers table now has two new columns called Longitude and Latitude. For each record these columns contain the precise location of the zip code centroid for the zip code area shown in the ZIPs drawing that has the same ZIPCODE value as the ZIP value for that record. In the illustration we’ve formatted the columns so that only four digits appear after the decimal point (see Formatting Columns).

If we like, we can create a drawing from our newly geocoded table. To do so, in the project pane we Copy the table and then Paste As a drawing.
In the Paste As Drawing dialog that appears we choose the Name and the Zip fields and choose the Longitude and Latitude fields to be used as our X and Y fields. These fields will be loaded into the X and Y boxes by default since Manifold knows that table columns called "Longitude" and "Latitude" quite likely contain the desired values. This also illustrates the wisdom of using these names in the Match Drawing dialog.

The result of the Paste As operation is to create a new drawing in the project called Customers 2 by default (since we already have one component, the table, called Customers Manifold will call the next component created by pasting it "Customers 2").

We can drag and drop the new Customers 2 drawing into a map that also shows the ZIPs drawing. This drawing shows us the points at which each customer record in the table was geocoded. We can use this map to discuss three important aspects of geocoding using Match.

First, note that locations taken from the ZIPs drawing objects appear at the centroids of those objects. With unusual, bow-shaped areas (such as those at the bottom center of the map) the centroid of an area will often be located outside of the area.

Second, if we have more than one customer at the same zip code there will be more than one point created at exactly the same location. If we wish to disperse these points slightly we can use a script or other method to disperse them or to create displays that show the density of customers at different locations.
Third, if the drawing used to geocode the table has more than one object with the same target value, only the first such object encountered will be used to assign a location to the record. For example, suppose we had more than one area in the ZIPs drawing with the same ZIP code. Any record with that ZIP code would take its location from the first such area encountered (by Object ID order). This is not an issue with full five digit zip code areas like ZCTAs (in which every five digit numeric code is guaranteed to be unique), but it can be an issue when geocoding to other entities.

For example, suppose we use Match to geocode records to a drawing of US Counties. Most such drawings have many counties that consist of several area objects; for example, in the case of coastal counties that have a mainland area as well as several other areas showing coastal islands that are part of the county. Each such area object will usually have the county name, FIPS code and other fields with the same value. If we matched a table of records to such a county drawing the centroids will be taken from the first object with the given matching field value. If we are matching using FIPS code the first object with a particular FIPS code could easily turn out to be a coastal island for some counties. In that case, the point for the matched record would appear at the island and not on the "main body" of the county area where we intuitively might prefer to see it.

That may well not matter in some cases. Match is often used in tasks where the objective is not exact spatial positioning but rather to see a representative spatial distribution where one's customers (or other records) are located. In the example above, for instance, it may well be that we don't really care so much exactly where the customer is located but rather just wish to know if our customer base is located mostly in the South Bay or in the North Bay. If the latter interest is our objective we really won't care whether or not a particular record is geocoded to an "island" bit of an area instead of the "main" area.

However, if we really do care whether or not our table is geocoded to an "island" bit of an area, we can take a moment to clean up the drawing being used to guide the Match so that it contains only one area for each value used to geocode. In the case of a counties drawing where some counties consist of multiple areas we have many ways of undertaking such a clean up.

For example, we could use the Union transform operator to create a single area from all areas having the same FIPS code. Another approach might be to use the Area (I) intrinsic field within a query to find all areas smaller than a given size (the islands, presumably) and to move them to a separate drawing. In that case they could still appear in maps for visual fidelity but would not participate in analytic operations such as Match using the drawing.

**Tech Tips**

When working with ZIP codes or postal codes make sure to use text columns for the zip code or postal code. Many postal codes, such as ZIP codes, have leading "0" characters (with ZIP codes such as "02138") that are significant and will be preserved if a text column is used and lost if a numeric column is used.

If we have the Manifold Geocoding Data option installed we can use Street Address Geocoding to geocode a table of ZIP codes. In that case, there is no need to use Match.

**See Also**

Zip Codes are Not Areas

**Street Address Geocoding**
Decision Support System

The Manifold Decision Support System (DSS) provides a way to select records using flexible criteria. DSS results are reported as percentage values from 0 to 100 percent within a Rank Column in a table. We use the Add Rank Column Dialog to add rank columns by specifying DSS criteria and a DSS query they should use.

Instead of using fixed criteria such as “greater than 50” in a query, DSS can handle a situation where the criterion might be more than 50 most of the time but sometimes a value near 50 is also acceptable. The science of using flexible criteria is generally referred to as “fuzzy” logic. See the Introduction to Decision Support topic for more information on fuzzy logic.

Decision Support describes objectives for each data field using criteria. Criteria are curves that show which values for the data field are more desired and which are less desired. One or more criteria may then be used in a ranking query to find records that are a good fit to the combined requirements of all the criteria used. The results will be reported using a rank number, a percentage from 0 to 100 percent that appears within a Rank Column in the table. Records that fit the query well will be ranked at or near 100 percent. Records that fit the query less well will be ranked at increasingly lower percentages.

Records may then be organized by clicking on the Rank Column heading to sort the table in order of DSS ranking.

To select records using Decision Support:

1. Open the table of interest.
2. Use Table - Add - Rank Column to add a Rank Column to the table.
3. In the Add Rank Column dialog, specify the name of the column.
4. Press New to add a new criterion to the Criteria pane.
5. In the Add Criterion dialog, choose the name for the criterion, the field (column) to be used and the type of criterion. Types such as High and Low provide presets that are automatically computed based on the data in the table. Press Apply to see the effect of the Type and press OK to create the criterion.
6. Continue adding criteria using steps 4 and 5 above for all the fields you wish to use. Criteria appear organized in a hierarchical diagram under the names of the fields they use.
7. Create a query by dragging and dropping criteria from the upper pane into the lower pane. Change the Junction from Or to And as desired by double clicking onto the Or. Place a Not in front of a criterion if desired by double clicking into the Not column and choosing Not.
8. Press OK and a Rank Column giving DSS values for all the records will appear in the table. Sort the column by clicking on the column head to see the records displayed in rank order. Records with a rank value of 100% most closely match the rank query.

Another way to make a selection using DSS is to use a ViewBot in the ViewBots pane. For example, suppose we have created a Rank Column named Ranking. We could add a ViewBot that reported for the Ranking field the records Greater or Equal To .90. This would report all records where the total Ranking was greater than 90%. We could then click Replace Selection in the ViewBots pane to select those records. Note that we use a value of .90 in the ViewBot since rankings are percentages.

Decision Support System Topics

Introduction to Decision Support - A quick and easy introduction to Decision Support.
Add Rank Column Dialog - Overview of all controls and panels in the main dialog used to create rank columns and to build criteria and queries reported in rank columns.

Take a moment to read the above topics in order to get the most out of Decision Support System.
Introduction to Decision Support

Much has been written about "fuzzy logic" in recent years. The basic idea behind fuzzy logic is to replace inflexible search criteria with criteria that have some give and take. Searches in real life often incorporate gray areas or subtle trade-offs that are not well represented by all-or-nothing selection parameters.

For example, suppose we are interested in finding automobile drivers who are "mature" in age. A traditional approach using SQL might be to select all records with an \texttt{Age} field value over 35. The problem with the traditional approach is that a driver of age 34 would never be selected even though such a driver when all other criteria are reckoned might well be considered as mature as a driver of age 35 or 36. Traditional systems are not well suited to making selections when one seeks flexible criteria such as "mature", since there is no magic age number at which (suddenly) 100% of the population becomes mature, with 100% of the population having been "immature" before that age.

In contrast, a fuzzy query might select records over 35, but would also allow some blurring of the criteria so that some records that are almost over 35 have a chance of being selected. At times (depending on other selection criteria) some records that are slightly over 35 will not be selected. A fuzzy selection criterion could define "mature" so that from age 20 on people are more likely to be considered "mature" with almost everyone by age 50 considered to be "mature."

Curves Show the Desirability of Different Values

How do fuzzy logic systems know when it is acceptable to use values that are close to 35 and when it is not acceptable? This turns out to be surprisingly easy to accomplish. We specify how likely values on either side of 35 are to be selected by specifying a weight or preference factor for their desirability. In Manifold's Decision Support System, we use curves to specify a criterion that shows the desirability (from 0 to 100 percent) of a particular value.

Since many people don't always think in curves or like to draw curves, Decision Support has several methods that make it easy to specify a criterion without having to draw a curve. For example, Decision Support provides "presets" that make it easy to pick common desirability criteria (such as \texttt{High}, \texttt{Low}, or \texttt{Average}) by clicking on a menu choice. Another method provided is the use of simple confidence intervals in the \texttt{Auto} method that Decision Support automatically translates into appropriate curves.

Let's take a look at how simple curves can show the desirability of different values. The following illustrations show grids and numeric values to help teach the concepts. The \textit{Add Criterion} dialog in Manifold used to add desirability curves is simpler and does not show grids labeled with numeric values.

![Desirability Curve](image)

The curve above shows an "all or nothing" choice of 35. Anything below 35 has zero interest, and anything above 35 has 100 percent desirability. Curves like this are called \textit{step functions}, for obvious reasons. Traditional SQL and other query systems use such all-or-nothing step functions.
The curve above shows a classic flexible approach to picking values at 35 and above. It’s been drawn so that desirability rapidly increases from 0% to 100% in the range from 32 to 37. Instead of an all or nothing step function that changes at 35, the numbers just above 35 are more desirable, but not absolutely so, than the numbers that are slightly lower than 35. Numbers that are well below 35 are not at all desirable while numbers far above 35 are 100% desirable.

The third curve shows a situation where there is a very gradual increase in desirability. It would be used in a case where one is very flexible about using values on either side of 35. In fact, depending on what one is searching for it might be a good criterion to use for age when seeking “mature” age groups. This might be a good criterion for “mature” age selection for marketing certain travel packages or upscale goods such as luxury full-sized automobiles.

Manifold’s Decision Support System combines criteria into queries where the desirability of various field values implied by the criterion are mathematically combined in a composite ranking of records. For example, suppose the values in the illustrations above represent the age of automobile drivers. Suppose for marketing purposes we wish to find all customers in our table who are mature drivers that live in an affluent area. We could begin by using the third criterion illustrated above for the Age field of our table under the name “mature”.

We then could create another criterion such as the one shown above using a Household Income field. In this case, we’ve drawn a criterion curve that increases rapidly for incomes around 40000, a level of household income well above average. We might give this criterion the name “affluent”.

To find regions with people who are both mature and affluent, we would use the above two criteria in a query that searched for mature AND affluent people. Decision Support System would spring into action on our behalf to sift through the entire table comparing the age and household income values of each record both to the mature
criterion as well as to the affluent criterion and ranking each record based on the composite desirability of the values.

For example, a person aged 54 with an income of 49000 would have 100% desirability in both criteria and thus would be ranked high. A person who is aged 54 but has an income of 40000 would have 100% desirability in the mature criterion but only about 70% desirability in the affluent criterion and so would receive a lower ranking. By mathematically combining the results of comparing individual records to the various criteria used in the query, Manifold Decision Support can create a composite ranking even within very complex queries. The result is a selection that incorporates our preferences in "gray areas" where inflexible selection rules cannot.

Using Decision Support System

The actual mechanism for using Decision Support System within Manifold is very simple: we add a rank column to the table that reports the relative ranking of records using the desired criteria in the desired way. We add a rank column with the Add Rank Column dialog, which also lets us specify criteria and their use within a DSS query.

See the Add Rank Column Dialog for details on adding rank columns to tables.
Add Rank Column Dialog

The Table - Add - Rank Column menu choice launches the Add Rank Column dialog. This dialog is used to create and manage criteria and Decision Support System (DSS) queries to be used with a table. The Add Rank Column dialog also appears when a rank column head is right clicked and Edit is selected from the context menu. It contains a criteria pane and a rank query pane. To add a rank column to a table we specify criteria and then use those criteria to create a ranking query. When we press OK the column will appear in the table with values computed by DSS.

Name
The name of the rank column as it will appear in the table.

Criteria
A hierarchical diagram of each field used in a criterion and the names of criteria created using that field. The illustration shows we have two criteria created: one using the Quantity field and one using the Unit Price field. The Criteria pane will show all of the criteria that have been created in this project for use in any rank column.

New - Add a new criterion.
Delete - Delete the highlighted criterion.
Properties - Edit the highlighted criterion.

Query
The lower pane shows queries created by dragging and dropping criteria from the upper pane to the lower pane. The query in the illustration finds all records that are Rare and very Expensive.

Not
Double click to prefix criterion with Not, for example, to create "Not Rare and …"

Hedge
Adjust the criterion by emphasizing or de-emphasizing its action. Double click into the cell to change. The illustration above shows the use of the hedge very to modify Expensive.

Criteria
Criterion dragged and dropped from the upper pane into
the query.

**Junction** Boolean **Or** or **And**. By default, criteria dropped into the query use **Or**. Double-click to change to **And** if desired.

### The Criteria Pane

The **Criteria** pane in the **Add Rank Column** shows all criteria that have been created in the project for use with any rank column. Criteria are shown organized under the field they use. Criteria can be used in queries as set forth below. A criterion may be used in a query in more than one rank column.

### Add Criterion Dialog

Clicking on **New** in the **Add Rank Column** pane launches the **Add Criterion** dialog. The dialog shows a curve in red with a vertical gray line at the **Center** value position and labeled with that value. The line along the bottom is labeled with the lowest and highest values found in the **Column** being used.

<table>
<thead>
<tr>
<th>Name</th>
<th>Name to use for this criterion.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Column</td>
<td>Field to use for this criterion. Criteria can only be created using numeric fields that are not Active Columns or other rank columns. The graphical display is labeled with the lowest and highest values found for the <strong>Column</strong> being used.</td>
</tr>
<tr>
<td>Preset</td>
<td>The type of criterion curve to create. Choose from <strong>Interval</strong>, <strong>Average</strong>, <strong>High</strong> or <strong>Low</strong> presets.</td>
</tr>
<tr>
<td>Center</td>
<td>Typically used with <strong>Average</strong> to specify the center of the bell curve. Also used to shift the center of the transition in <strong>High</strong> or <strong>Low</strong> presets. Check <strong>Auto</strong> to have Manifold reckon the center automatically. A gray line shows the value for <strong>Center</strong> with the value labeled.</td>
</tr>
<tr>
<td>Within</td>
<td>Typically used with <strong>Average</strong> to specify the width of the bell curve. Also used to widen or narrow the transition zone from high to low in <strong>High</strong> and <strong>Low</strong> presets. Check <strong>Auto</strong> to have Manifold reckon the width automatically.</td>
</tr>
<tr>
<td>(Interval Boxes)</td>
<td>Appear with <strong>Interval</strong> preset. Choose those values that bracket the given percentage above or below the region of transition. See discussion below.</td>
</tr>
</tbody>
</table>

The first step in running Decision Support System is to add one or more **criteria** to the Criteria list. Each criterion specifies the desirability (from 0 to 100 percent) of various possible values in one field. For example, we might create a criterion called "Mature" that specifies the desirability of various ages taken from a [Median Age] field in our table. We might create another criterion called "Affluent" that showed the desirability of various values taken from the [Household Median Income] field in our table. Each criterion describes the values we seek within a single field. We create criteria with the intention of using them in queries.
More than one criterion can be created for each field. For example, suppose we wish to characterize different patterns of median age in several different ways. We could create three different criteria, called "Mature", "Young" and "Old" that are all based on the [Median Age] field. We could then use whichever criterion was desired to rapidly customize a query. For example, we could rapidly switch from seeking areas that are "Mature" and "Affluent" to finding areas that are "Young" and "Affluent".

All criteria that appear in any rank column dialog in a table are available to all other rank column dialogs in the table. We might therefore create several different criteria using the same field and use them in several different rank columns in that table.

To Add a Criterion:

1. Press New to add a new criterion to the Criteria pane.
2. Enter a name for this criterion in the Name box.
3. Choose the field (column) from the table that will be used in this criterion.
4. Choose a preset type of criterion.
5. Make any manual changes desired using Center, Within or interval boxes. If any manual changes to Center or Within or interval boxes have been made, press Apply to apply these changes to the criterion curve.
6. Press OK.

Methods used to specify Criteria

Every criterion will be shown as a red curve in the criterion diagram. There are several methods that may be used to specify a criterion curve:

- **Use a preset** - Choosing an Average, High or Low method is the easiest approach because the system does all the work.
- **Customize a preset** - Each of the presets allows alterations, if desired, in the defaults picked by the system by un-checking the Auto boxes and then manually changing the values in the Center or Within boxes. For example, we might start with the Average preset but choose a wider bell curve by specifying a larger value in the Within box so that a wider range of values are more desirable. This is almost as easy as using a preset.
- **Upper and Lower confidence interval ranges** - Using the Interval preset, we can pick upper and lower confidence ranges and have the system figure out an appropriate criterion curve. This is good for situations where we can pick an approximate upper and lower bound for what we are seeking.

Standard deviation is used create some of the presets based on the actual distribution of values within the table. "Standard deviation" is a statistics measure that measures how much numbers are scattered about their average. If the numbers follow a perfect Gaussian ("bell shaped") distribution, than 68% of the numbers will be within one standard deviation +/- from the average and 95% of the numbers will be within two standard deviations of the average.

**Average**

The Average preset creates a bell curve that is Centered on the average value in the data field and which extends one standard deviation to either side of the average. We can customize the default preset by changing the Center to change the center line of the bell curve or by changing the width of the bell curve. We may change the width of the bell curve by changing the Within number. Press Apply after changing the values.

Why would we change the Center? Perhaps we want to create a criterion that we will name "Better than Average." To create this criterion we would like to have a bell-shaped curve but centered about a higher point than the average.

**High and Low**

We often wish to search using the "high" and "low" values of a data field. For example, seeking "Affluent" regions usually means find those areas with "high" household median income. Manifold Decision Support System provides High and Low presets that make it easy to create a criterion that finds above average or below average values for a data field.
• The **High** preset assigns zero desirability to all numbers below the average and then increases desirability in a smooth ramp to 100 percent for numbers that are greater than one standard deviation above the average.

• The **Low** preset assigns 100 percent desirability for numbers up to one standard deviation below the average, and then decreases desirability in a smooth ramp to zero for numbers above the standard deviation.

Both **High** and **Low** presets may be customized by changing **Center** or **Within** values.

**Interval**

The **Interval** method of specifying a criterion uses two desirability intervals (confidence values) to pin down a curve from zero desirability to 100 percent.

The default **Interval** curve uses a number that is 10% desired and a number that is 90% desired and draws a curve between them. The numbers in the boxes to the right (the 10% and the 90%) are desirability as a percent. The numbers in the left-hand boxes are data values for the data field being profiled. To use **Interval**, change the numbers given for what is 10% desired and what is 90% desired and then press **Apply**.

For example, if we were searching for values and knew that a number of about 15 was 10% desired and a number of 45 was 90% desired, we would choose **Interval** as the type of criterion and then use 15 = 10% and 45 = 90%.

Suppose we weren’t sure what was 10% desired, but we knew very well that 25 was 80% desired and 45 was 90% desired: we would enter 25 = 80% and 45 = 90%.

**Names**

When creating a criterion it helps to provide a descriptive, memorable name for the criterion. For example, perhaps a certain pattern of income characterizes people our marketing department calls “Boomers,” in reference to a particular demographic cohort arising from the post-World War II “baby boom”. If we add a criterion based on [Household Median Income] that matches this income pattern we might refer to that criterion as a “Boomer” criterion.

**Creating Queries for Ranking**

![Add Rank Column](Add%20Rank%20Column.png)
The lower pane in the Add Rank Column dialog hosts queries. The query in the illustration finds all records that are Rare and very Expensive. Queries are created by dragging and dropping criteria from the criteria pane into the query pane. As each criterion is dropped into the query it is added to previous criteria using Or by default.

- **Not**: Double click into the Not column to prefix criterion with Not, if desired.
- **Hedge**: Adjust the criterion by emphasizing or de-emphasizing its action. Double click into the Hedge column to specify, of desired.
- **Criteria**: Criterion dragged and dropped from the upper pane into the query.
- **Junction**: Boolean Or or And. By default, criteria dropped into the query use Or. Double-click to change.

**Boolean Combinations of Criteria in Queries**

"Boolean" refers to logical operators such as AND, OR and so forth. Suppose we have a query similar to the one in the illustration above. It includes two lines using criteria: Rare and Expensive with a junction of And. In addition, we could double click into the Not column to create constructions such as "Not Rare and Not very Expensive." Various combinations of the Junction and Not columns result in Boolean combinations such as And, Or, And Not ("NAND" in Boolean terminology) and Or Not ("NOR").

There can be as many query lines as we wish with each line after the first ending with a Boolean operator that shows the role it is to play in the query and the next line optionally preceded with a Not. The order of precedence is simply from first line to last. Decision Support System uses simple queries that do not use nested sub-expressions within parentheses. It is possible that complex queries will result in zero values for all records, meaning that no record comes close to the query criteria.

For a historical note on George Boole, the inventor of Boolean logic, see the Expressions topic.

**Hedges**

Suppose after creating a criterion such as Expensive we would like to use this criterion in a query searching for records where products are extremely expensive? One approach might be to create another criterion using the High preset and then use the Center box to shift the criterion curve towards higher values of unit price. Another method is to Double-click into the Hedge column in the query pane to choose extremely.

This will change how the criterion is used in the query without any need to go back and change the criterion. When we press OK the Expensive criterion will automatically be shifted so the rank column will be recomputed using extremely expensive.

**Hedge** choices provide standard modifications to the use of criteria as follows:

- **(blank)** Applies criterion "as is".
- **above**: Favors values above the last maximum of the criterion. Applying this hedge to an Average criterion yields a High
criterion with the same settings.

**below**  Favors values below the first minimum of the criterion. Applying this hedge to an Average criterion yields a Low criterion with the same settings.

**extremely**  Greatly strengthens the criteria defined by the criterion. If the original criterion rates a value of 250 dollars with a 50 % desirability level, applying extremely will rate the same 250 dollars a 79.4 % desirability level (79.4 % = 50 % ^ 1/3).

**slightly**  Greatly weakens the criteria defined by the criterion (weaker criteria than somewhat). In the example above, applying slightly will rate 250 dollars with a 12.5 % desirability level (12.5 % = 50 % ^ 3).

**somewhat**  Weakens the criteria defined by the criterion, but not as weak as slightly. In the example above applying somewhat will rate 250 dollars with a 25 % desirability level (25 % = 50 % ^ 2).

**very**  Strengthens the criteria defined by the criterion, but not as strong as Is Extremely. In the example above applying Is Very will rate 250 dollars with a 70 % desirability level (70 % = 50 % ^ 1/2).

---

**Deleting Criteria**

When deleting a criterion, keep in mind that it may be used in other rank columns than the one being edited. Deleting a criterion will remove it from all queries in which it participated. This can change the meaning of a DSS query used to generate a rank column.

For example, suppose we have a criterion called Populated that finds all records where a population field is High. Suppose we also have a criterion called Small that finds all records where area is Low. Suppose we have created a rank column called Crowded that uses a query with both of these criteria to find records that are Populated and Small.

If we open some other rank column for editing and then delete the Small criterion without realizing that it is used in the Crowded rank column, we will change the meaning of that rank column. When the Small criterion is deleted, what is left of the query for the Crowded column will simply find records that are Populated.

If a rank column uses only one criterion and that criterion is deleted, the rank column will be deleted as well.

**See Also**

Decision Support System - for more detailed topics on using DSS in Rank Columns
Introduction to Decision Support - for more information on fuzzy logic.
Rank Columns / Decision Support System - a step-by-step example of creating a rank column and using DSS.
Charts

Charts
Manifold provides two pathways to charting data:

- **Charts** are project components created from database tables. Manifold's Chart system provides a simple bar chart style that shows data from tables in 2D charts.
- **Minicharts** are small, simple charts that hover above objects in a drawing. They are kept deliberately small and simple since there are often many of them in view in a drawing or map.

To Create a Chart from a Table

1. Choose **File - Create - Chart**.
2. In the **Table** box choose the table that will be used to create the chart.
3. In the **Labels** box choose the field that will be used for the X-axis in the chart. To group records and present them on the X-axis choose the grouping field in the **Group by** box. This will override the **Labels** box.
4. In the **Display** pane choose the fields that will be plotted on the Y-axis in the chart.
5. Press **OK**. The chart will appear in the project pane under the table from which it is created.

Create Chart Dialog Controls

- **Name**: Specify a name for the chart component in the project.
- **Table**: The table to be used to create the chart.
- **Group by**: Field to use to group displayed values. If a **Group by** field is specified it will be used to order the results on the X axis and will override any **Labels** setting.
- **Labels**: Field to use to arrange results on the X axis.
- **Display**: Choose one or more fields to display on the Y axis.
- **Select All**: Check all columns. All data fields will be charted.
- **Select None**: Uncheck all fields. No data will be charted.
- **Select Inverse**: Use this to uncheck all fields before checking those desired.
- **Move to Top**: Move the highlighted field to the top of the field stack. If several fields are displayed in the chart, this displays it as the leftmost bar within a group of bars.
- **Move Up**: Move the highlighted field up one position in the field stack. This displays it one position to
the left within a group of bars.

- **Move Down** Move the highlighted field down one position in the field stack. This displays it one column position to the right within a group of bars.

- **Move to Bottom** Move the highlighted field to the bottom of the field stack. This displays it as the rightmost column within a group of bars.

- **View Types** Show a column displaying database types for each field.

**Numeric and Non-Numeric Fields**

Numeric fields chosen for display will be shown using their numeric value in the Y axis. Non-numeric fields chosen for display will be counted and the total count in each grouping will be shown in the Y axis. For example, suppose we have a table that shows customer address records. The table includes a **City** field and a **State** field, both of which are text fields. If we selected **State** as the **Group by** field and in the display pane checked **City**, the chart would show how many cities are in each state.

**Changing Colors**

Colors in charts will be assigned using a default palette. Choose **View - Format** to change colors. Colors may be changed for each individual field displayed by double clicking into the color well shown for that field. Colors may also be changed for all fields at once by loading a different palette. Colors will be the same in all windows showing that chart.

- **Load Palette** Load a new palette from the collection of palettes available within Manifold.

- **Interpolate** Adjust colors in a smooth gradation from the upper color to the lower color.

- **Reverse** Reverse the order of colors used.

The **Load Palette**, **Interpolate** and **Reverse** commands are normally used only when several fields are charted since it is faster to change colors with only one field by double-clicking into the color well.

**Labels**

Choose **View - Labels** to place labels on the X axis or on the Y axis. Reasonably round numbers will be chosen for the Y axis labels based on the values of the data displayed. X axis labels will be drawn from the **Labels** field or from the **Group by** field, if one is specified.

X axis labels may be rotated vertically to allow more labels to fit on the X axis. Labels will be clipped to avoid overlaps. Therefore, every cluster of bars might not be labeled.

Charts will automatically expand as the chart window is resized. To see more labels, make the chart wider. On low resolution monitors, undock or close the project pane and other panes if need be to allow the full width of the screen to be used.

**Logarithmic Scaling**
Charts

Click **View - Logarithmic Scale** to use logarithmic values on the Y axis. This will re-scale the Y axis using values such as \(10, 100, 1000, 10000\) and is a good way to show two different series of values where one series has relatively small values and the other has relatively large values.

**Changing Fields Displayed**

Once a chart is created we can choose **Chart - Data** to change the fields displayed or the **Group by** or **Labels** fields used.

**Example**

Import `mexico_eg.mif`, the sample drawing of Mexican provinces. Use **File - Create - Chart** to launch the **Create Chart** dialog. Choose `Mexico_eg Table` as the table to use. Group by **Place_name**. In the **Display** pane check the boxes for **BUSES_1991** and **TOTVEH_199**. Press **OK**.

A new chart component appears in the project pane under `Mexico_eg Table`. We can double click on the chart to open it. Note that labels are arranged on the X axis to avoid overlaps, so that not all bars are labeled.

We can choose **View - Labels** and check the box to rotate the X labels vertically. This allows more labels to be visible on the X axis.
If we want more labels to appear, we can widen the chart window to allow room for more labels. Some of the bars are very short because there are relatively few buses in comparison to total vehicles.

We can remedy this by choosing View - Logarithmic Scale so that the bars representing buses become more visible.

**Non-Numeric Example**

Import the Customers table from the Nwind.mdb sample database (the same as the Northwind.mdb sample database shipped with Microsoft Access).

<table>
<thead>
<tr>
<th>Customers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Custo...</td>
</tr>
<tr>
<td>AUPK1</td>
</tr>
<tr>
<td>ANATH</td>
</tr>
<tr>
<td>ANTON</td>
</tr>
<tr>
<td>ARORH</td>
</tr>
<tr>
<td>BERGS</td>
</tr>
<tr>
<td>BLAUS</td>
</tr>
<tr>
<td>BROWD</td>
</tr>
</tbody>
</table>

We would like to see what is the most frequently occurring customer title in our database. To find out, we create a chart from this table using **Contact Title** as the Group by field and use **Contact Name** as the Display field.
The chart shows a bar for each **Contact Title** that exists. The height of the bar is set by how many different records exist with a value in **Contact Name** that also have that particular **Contact Title**. We can see the "Owner" and "Sales Representative" are the most frequently occurring titles.

The **Contact Name** field is used as a counting field. We could have used any field where we know there is a value in the field whenever there is a value in the **Contact Title** field. Looking at the table it seems reasonable that every record that has a value in the **Contact Title** field will also have a value in the **Contact Name** field so we used that field as a counting field.

**Saving Charts as Images**

The Tools - Make Image command captures the current scene displayed in a chart window as an image. Images made of chart windows are captured at screen resolution.
Minicharts
Minicharts are small charts that hover above objects in a drawing. They are frequently used with drawings showing areas, but they may also be used within drawings that show points.

They display the relative values for each object of specified numeric data fields from tables associated with the drawing.

To Create Minicharts in a Drawing

1. Click open a drawing.
2. Choose Drawing - Minichart
3. In the Minicharts dialog, check the field boxes for numeric fields that are to participate in the minichart.
4. Double click into color wells to change colors from defaults, if desired.
5. Choose the display Style, Size and Frame color and other parameters. Press OK.

To Remove Minicharts from a Drawing

1. Open the drawing.
2. Choose Drawing - Minichart
3. In the Minicharts dialog click the Select None button to uncheck all fields. Press OK.

The minichart dialog appears with a range of colors for each numeric field that may be used to create a minichart. Double click into any well to change the color.

Minichart Dialog Controls

- **Select All** - Check all fields in the Chart pane.
- **Select None** - Uncheck all fields in the Chart pane.
- **Select Inverse** - Uncheck all checked fields and check all unchecked fields. A fast way to show all but one field: click Select None, check the one column not desired and then click Select Inverse.
- **Move to Top** - Move highlighted field to the top of the list.
- **Move Up** - Move highlighted field up one position in the list.
- **Move Down** - Move highlighted field down one position in the list.
**Move to Bottom** - Move highlighted field to the bottom of the list.

**Style**
Choose from common types of simple charts.

**Frame color**
Color to use for frame border or other outlining element in the minichart.

**Offset point charts by**
When a minichart is created based on data from point objects, specifies the number of printer's points in the X and Y directions to offset it from the point. By default, the offset is 6 points in both X and Y. Increase these values to position the minichart farther away from the point. Offsets must be in the range from -100 to 100 points.

**Size**
Size of the minichart in points.

**Resolve overlaps**
Checked by default. Do not show minicharts that overlap each other. Uncheck to show all minicharts.

**Spacing**
Minimum spacing between minicharts in points before minicharts will be hidden to resolve overlaps.

**Use with Restraint**

Minicharts are visually pretty and fun to use but they should be used with restraint. When used with too many fields they may cause confusion instead of enlightenment. When used tastefully with a few fields they can convey information that is otherwise difficult to show.

Readers interested in creating clear and effective maps and charts will probably enjoy the spectacular series of books on this theme by Edward Tufte, beginning with "The Visual Display of Quantitative Information", ISBN 0-9613921-0-X. This is a fabulous book that is light on text, easy to read and full of wonderful illustrations.
Surfaces and Terrains

Surfaces are components similar to images in that they contain data organized as pixels. Surfaces are made up of numbers arranged in a regular array. Surfaces are imported from formats in which the data are organized in regular form, usually in a raster format similar to images. Surfaces differ from images in that whereas images save color or grayscale in the regularly organized data, the numbers in a surface normally will store elevations, temperatures or other abstract data.

Surfaces may also have Terrains created for them. A terrain is a 3D view of a surface. When a surface is imported a default terrain is created for that surface. Additional terrains may be created for a surface and each such terrain can have its own coloring and other characteristics. Creating an additional terrain for a surface does not require any more storage space since terrains are just different views into the one data set contained in the surface. An unlimited number of terrains can be created for each surface. If desired, the default terrain created for a surface may be removed so that the surface has no terrains.

Because surfaces are structurally similar to images they are imported into Manifold as components that have many characteristics that are similar to images. Surfaces will have commands available that are very similar to images. In particular, one can:

- Make selections in surfaces as in images using the pixels in surfaces.
- Modify selections using Edit - Modify Selections.
- Save and Load masks as with images.
- Use many transform operators similar to those used with images.
- Add a surface to a map as a layer.
- Georegister a surface.
- Print a surface.
- Analyze a surface using Surface Tools or the Volume function in SQL.

By default, when a surface is opened it appears in a 2D window like an image.
The default settings used with surfaces provide a 2D visual image of the data set contained in the surface. By default, images are shown shaded with autocontrast. By changing Display Options they can change their appearance, for example to show simple grayscale or to be colored with a palette like the first illustration in this topic.

The illustration above shows part of a DTED Level 2 terrain elevation surface from NIMA that is overlaid in a map with hydrology layers from the analogous VMAP Level 2 vector data set. The surface Surfaces may be used within maps as layers by dragging and dropping the surface into the map. Surfaces may also be copied and pasted as images, and those images can then be used as layers in a map as well.

When a surface is imported into Manifold, a Terrain for that surface is automatically created and placed in the project pane hierarchy under the parent surface. Terrains are components that provide a 3D view of a surface.
Data sets imported as surfaces usually are elevation data showing the height above sea level for various regions. Typical formats used for such data are NIMA's DTED or various DEM formats. However, we may use any regularly organized X,Y data set containing a third numeric value to be used as an "elevation." In general, any raster data set can be used as a surface.

Surfaces can be used for displaying terrain surfaces as well as for showing abstract data in gradient maps. See the Displaying Data in a Gradient Map topic for an example that shows demographic data displayed in a continuously varying gradient throughout the United States using a surface.

Some formats can be imported into Manifold either as an image (with File - Import - Image) or as a surface (with File - Import - Surface). When a surface data set is imported, a 2D image component will be created as well as a 3D terrain component.

We import a surface using File - Import - Surface. Manifold's importers can handle a wide array of formats with some overlap between the surface format importers and the image format importers. A few formats, such as USGS Digital Elevation Modules (DEM) can make sense either as surfaces, drawings (a field of points arranged in a regular grid), or images (with a palette applied to show a 2D image of elevations). Which component type is used for such data depends on its intended usage within the project. See the Import and Export topic for more information on specific import dialogs used to import surfaces.

**Surface Windows**

When a surface is opened in a surface window or appears as a layer in a map, the appearance of the surface is controlled by options in the View - Display Options dialog. By default, the surface appears as a grayscale image with shading and autocontrast. These options are suitable for images that represent terrain elevations. They may be modified to better show surfaces that represent abstract data, such as land use codes.

Note that although surfaces appear in windows like images and share many commands that may also be applied to images, they are not images. The appearance of the data within the surface window is controlled by the display options, some of which (such as slope or aspect) may involve computation. In contrast, the appearance of images within an image window is "hard wired" by the color values of the pixels.

If desired, a surface may be converted into an image by copying the surface and pasting it as an image. In that case, the colors in the image pixels will be taken from the display options currently set for the surface.
The image above shows the bathymetry of the Northern half of Lake Tahoe on the Nevada-California border in the United States. It was imported from an .e00 grid file as a surface and then copied and pasted as a detailed, 1995 x 3416 pixel image. The Invert command was then used to invert the grayscale in the image to give it a greater sense of depth.

Creating Surfaces

Surfaces are used for two main purposes:

- To show terrain elevations or bathymetry.
- To show abstract data such as population or temperatures as a continuously varying surface.

Surfaces showing terrain elevations or bathymetry are usually imported from pre-existing DEM or other surface formats since data sets for terrain elevations are usually so extensive that few users can create such data sets.

Surfaces showing abstract data are often created from drawings or tables. In this case, the drawing or table is copied and then pasted as a surface. The surface can then be colored with display options or seen in 3D as a "terrain." See the Creating Surfaces from Drawings and Tables topic for details.

Touch Selection in Surfaces

Touch selection in drawings will select the objects touched. In images, touch selection will select pixels within the tolerance setting in the Tool Properties pane. With surfaces the tolerance setting for touch selection is automatically set to be five percent of the overall height range. Clicking on a surface with height values that range from 1000 to 2000 at a point with a height of 1820 will select heights between 1770 and 1870.

Converting Data Types in Surfaces

Surfaces internally consist of a regular array of numbers. The numbers within surfaces can be any one of a variety of numeric types, either integers of various types (signed, unsigned, 8-bit, 16-bit, 32-bit or 64-bit) or single or double precision floating-point numbers.

The Surface - Convert To command allows conversion between different types of numbers within the surface.

Masks and Surfaces

Masks can be used with surfaces as they are with images. Saving a mask from a surface results in an image that is georegistered to the surface and that retains any selection that was made in the surface. This is a handy way of saving complex selections made in a surface as an alternative to using the Selections pane.

Using the Layers Pane with Surfaces
The Layers pane helps control the appearance of surfaces within surface windows. The layers pane includes checkboxes for two system "layers" - a background color layer and a border layer that shows an enclosing box about the height and width of the surface.

By default, surfaces are shown using the checkerboard background Manifold uses to provide a backdrop for any transparent regions. Checking the Border box in the layers pane will draw a one-pixel border around the height and width of the surface. This is a good way to see the actual extent of a surface that contains regions of invisible pixels.

Checking the Background box in the layers pane will replace the checkerboard background with whatever is the default background color. This is a good way to see the actual extent of a surface that contains regions of pixels that are the same color as the background (usually white).

Note that only maps can have true "layers" in Manifold in the sense that they can layer more than one component within the same map window. The border and background "layers" in the Layers pane for surfaces are not true layers even though they appear in the Layers pane in the same manner as do layers in maps. These are simply system controls that take advantage of the Layers pane as a conceptually convenient user interface.

**Layouts and the Layers Pane**

If a surface has any Layouts created they will appear as "layers" in the layers pane for that surface. Checking the box for one of these print layout layers will cause a layout rectangle to appear in the surface that shows the region covered by the layout.

Right clicking onto the hatched border of one of the layout rectangles in the surface will cause a context menu to appear with controls based on that layout rectangle. For example, we can **Zoom** to a given layout rectangle, **Print** it or change its **Properties**. If a layout is empty (for example, if the layout scope is set to **selection** and no pixels are selected in the parent surface) zooming to the layout will do nothing.

Use **Tools - Options - Colors - Layout Rectangle** to change the color in which layout rectangles are shown. The default color is black.

**Tech Tip**

The status bar will report the height of a surface at the cursor position when a surface window is open. When one or more surfaces appear in a map window, the status bar will report the height of the uppermost visible surface at the cursor's position.

**Notes**

See the Import a Raw Binary File - NLCD to see an example of a surface that shows data other than terrain elevation. The example shows a surface that represents data from the National Land Classification Data system.
Surface Window Menus and Controls

This topic lists default menus and controls that appear when a surface window is active. These will also appear when a surface tab is active in a map. Surface commands are all listed under the Surface menu. Some commands are organized under submenus called Adjust and Effects. These somewhat arbitrary categories like those used for images so Manifold users who are familiar with Adobe PhotoShop will find the menu structure more familiar when seeking analogous commands.

Edit Menu

**Undo**
- Undo last command.

**Cut**
- Windows clipboard cut operation. Copy pixels to the windows clipboard and delete them from the surface.

**Copy**
- Windows clipboard copy operation. Use this to copy pixels from the surface onto the clipboard for pasting elsewhere.

**Paste**
- Paste the contents of the Windows clipboard into the surface and deletes any selected pixels. Creates new pixels in the surface.

**Paste Append**
- Paste the contents of the Windows clipboard into the surface without deleting any selected pixels. Creates new pixels in the surface.

**Delete**
- Permanently delete selected pixels from the surface.

**Delete All**
- Permanently delete all pixels from the surface.

**Assign Projection**
- Used with drawing, image, labels and surface windows to change the interpretation of the data. This is a specialized function that is applied only when manually specifying the projection of a component imported from a format that does not correctly store projection information. To change the native projection of any component other than a map, use the Edit - Change Projection command.

**Change Projection**
- Used with components like drawings and images to re-project the component and thus permanently change the data.

**Select All**
- Select all pixels in the surface.

**Select None**
- Deselect all pixels in the surface.

**Select Inverse**
- Invert selection: those pixels that were selected will be deselected while those pixels that were not selected will be selected.

**Select Mode**
- Choose the selection mode to be used for mouse selection:
  - **Replace** - Any selection made with the mouse will replace the previous selection.
  - **Add** - Any selection made with the mouse will be added to the previous selection.
  - **Subtract** - Any selection made with the mouse will be subtracted from the previous selection.
  - **Invert** - Any selection made with the mouse will be inverted with the previous selection.
  - **Intersect** - Any selection made with the mouse will be intersected with the previous selection.

**Modify Selection**
- Changes the shape of regions of selected pixels.
  - **Border** - Select those pixels on the border of the selected region(s).
**Contracts and Terrains**

**Contract** - Uniformly contract selected region(s) by given number of pixels.

**Expand** - Uniformly expand selected region(s) by given number of pixels.

**Smooth** - Provide a smoother edge to the region(s) of selected pixels.

**Snap To**

**Snap To** Enable/Disable snap modes:

**Graticule** - Snap to graticule intersections if a graticule is displayed.

**Grid** - Snap to grid intersections if a grid is displayed.

**Save Mask/Channel**

Create masks from the current surface or save channels of the current appearance of the image as grayscale images.

**Load Mask/Channel**

Load a mask or a channel from a grayscale image.

**Open Data Source**

Opens in Database Console the data source of a surface linked from a table in an external data source.

**Go To**

Go to the entire surface or to a selection.

**View Menu**

**Back**

Go back one view.

**Forward**

Go forward one view.

**Zoom In/Out**

Zoom in brings us closer to the surface while zoom out moves us farther away.

**Zoom To Fit**

Zoom so that the surface fits the existing window size.

**Zoom To**

Zoom to a specified zoom level.

**Graticule**

Show a graticule (latitude/longitude lines).

**Grid**

Show a grid that may be used to measure or to align editing and snap commands.

**Legend**

Display a legend showing formats.

**North Arrow**

Display a North arrow.

**Scale Bar**

Display a scale bar.

**Display Options**

View and modify surface display options.

**Panes**

Call a dialog that allows displaying or hiding all panes.

**Full Screen**

Show the surface over the entire monitor area.

**Refresh Data**

Update a linked surface with the latest data from the controlling geocoded table.

**Refresh View**

Update the visual appearance of the surface with any changes made since the last refresh. Enabled only if Autorefresh View is off and some change (such as a selection) has been made that might affect the visual appearance of the surface.

**Autorefresh View**

Automatically update the surface whenever any changes are made that might affect the visual appearance of the surface. Turn this option off when working with very large surfaces where there is no desire to take the time to show, say, a selection when it is made (repainting a very large surface on every selection step might take longer than desired). When Autorefresh View is off, we
can always cause a refresh on demand using the Refresh View command.

Properties View the properties dialog for this surface.

View - Panes Menu Panes of special interest when working with surfaces include:

Selection Save and recall regions of selected pixels.

Surface Menu

Open Data Source Open Database Console focused on the data source of a linked surface.

Transform Transform surfaces using formulas. Available only when the Surface Tools extension has been installed.

Quantize Convert surface to the specified number of height values using the most frequently occurring heights.

Resize Change the number of pixels in the surface using desired interpolation method.

Convert To Convert the numeric types used for height information in the surface.

Contours Create contours using this surface.

Watersheds Compute streams, watersheds and upstream areas. Available only when the Surface Tools extension has been installed.

Relink Enabled for a linked surface only. Relinks (reconnects) the linked surface to another data source.

Unlink Enabled for a linked surface only. Converts the linked surface to a local surface with no connection to the originating data source. Linking and then unlinking a surface is equivalent to importing it.

Surface - Adjust Menu

Invert Invert pixel height value. Creates a topographic negative so that heights become low regions and vice versa.

Threshold Forces height pixels to maximum, minimum or unchanged based on a histogram.

Posterize Convert surface to a limited number of height values. Height values are chosen to be evenly spread throughout the total range of heights for the number of levels specified.

Surface - Effects Menu

Filter Applies convolution filters, either custom filters or a choice from numerous presets.

Noise Add random height values.

Tile Convert surface into rectangular tiled regions of appropriate height.

Tech Tip
The status bar will report the height of a surface at the cursor position when a surface window is open. When one or more surfaces appear in a map window, the status bar will report the height of the uppermost visible surface at the cursor's position.
Surface Display Options

Surfaces appear in a surface window as grayscale images where lighter colors indicate greater heights. The word "Height" is used even though the data values of a surface may convey some other meaning, such as temperature.

To change the appearance of a surface window, use View - Display Options to change the settings.

**Display Options**

By default, surfaces will appear as shaded surface renderings with the color of each pixel corresponding to the data value at that spot as adjusted for highlights and shadows from a Sun in the East. Other options include display of aspect or slope, or use of shading. Palette controls are similar to those used in Thematic Formatting of drawings.

**Display**
Choose the computation used to display the surface in 2D.

- **Height**: Color each surface pixel by its height.
- **Aspect**: Color each surface by its aspect, that is, the azimutal direction in which a tangent plane faces at that spot, for example, a north-facing section of hillside. Aspects vary from -180 to 180 degrees.
- **Slope**: Color each pixel by the inclination of the surface at that point. Slopes are given in degrees.

**Palette**

Preset color combinations that may be applied to surfaces. Press **Apply** to apply the palette to the surface. Palettes use a fixed number of intervals and corresponding colors.

**Apply**
Apply the chosen palette to the Colors pane. This allows scrolling through the palettes without changing colors until we press **Apply**. Pressing **Apply** only changes the color...
scheme in use for values. It does not change the formatting of the surface until the OK button is pressed. To see a preview of how the applied colors will look, use the Preview check box.

Reverse  Reverse the formats used in the colors box from high to low.

Interpolate  Change the colors or sizes used in the colors boxes by interpolating between the top and the bottom boxes. A quick way of creating smooth gradients of colors or sizes.

Lighten  Lighten all colors. Each click of the Lighten button lightens the colors a bit more.

Darken  Darken all colors. Each click of the Darken button darkens the colors a bit more.

Grayscale  Convert all colors to grayscale.

Load  Load a previously saved theme from an XML file. The loading rules allow omitting color specifications in the XML file to load values using whatever is the default color. Loading rules also allow omitting any of the colorDef, colorMax, colorMin, interval and type tags.

Save  Save this theme to an XML file.

Shading  Shade the surface with highlights and shadows as if illuminated by the Sun from a given azimuth and elevation.

Autocontrast  Automatically adjust contrast for even dispersion of light and dark tones.

Azimuth  The angle to the Sun’s position in degrees from North: North = 0, East = 90, South = 180 and West = 270 degrees. Keep in mind that in most latitudes in the Northern Hemisphere the Sun traverses azimuthal angles from East to South to West. Used to compute shading.

Altitude  The Sun’s altitude above the horizon in degrees. 0 is at sunrise or sunset and 90 is directly overhead. Used to compute shading.

Slopes are measured in degrees such that 0 degrees is a horizontal slope (an absolutely flat surface). 45 degrees is equivalent to a slope where for each meter in a horizontal direction the surface moves one meter vertically up. 90 degrees is an impossible example of a vertical slope. There are no slopes higher than 90 degrees. There are no negative slopes since every tilt of a surface can be viewed as either negative or positive and Manifold always regards it as positive.

Shading is computed based on a given position of the Sun as set by Azimuth and Altitude values. In the Northern Hemisphere in the course of a day the Sun moves in azimuth approximately from 90 degrees (East) to 180 degrees (South) to 270 degrees (West). Depending on the time of year and the latitude the Sun’s altitude will range from 0 degrees (on the horizon, at sunrise and sunset) to 90 degrees (directly overhead, in the tropics). To allow artistic effects and for usage in unusual latitudes (such as above the Arctic Circle), Manifold allows Sun azimuth and altitude values beyond these ranges.

Save Mask/Channel

The Edit - Save Mask/Channel dialog for surfaces includes choices to save aspect or slope data in single-precision floating-point format into new surfaces. Aspects vary from -180 to 180 degrees. Slopes vary from 0 to 100 percent. Surfaces created using this dialog will retain the coordinate system (projection) and saved selections of the original surface.

Example
We will use **File - Import - Surface** to import the *montara_mountain* DEM files from SDTS format.

This file came from the USGS free Internet servers. It shows a portion of the San Francisco peninsula approximately centered on the region where Highway 92 cuts across the San Andreas fault and the coast range on its way from the Bay to the ocean.

DEM's are "Digital Elevation Modules" and give the height of the Earth's surface. High resolution DEMs (1:24,000-scale) are provided by USGS in SDTS format. They may be imported from the **File - Import - Drawing** dialog or form the **File - Import - Surface** dialog. When an SDTS file containing a raster elevation data set is imported via the "drawing" dialogs, Manifold knows to create a surface.

DEM's are "Digital Elevation Modules" and give the height of the Earth's surface. High resolution DEMs (1:24,000-scale) are provided by USGS in SDTS format. They may be imported from the **File - Import - Drawing** dialog or form the **File - Import - Surface** dialog. When an SDTS file containing a raster elevation data set is imported via the "drawing" dialogs, Manifold knows to create a surface.

The result is several new components in the project: the surface, the surface's terrain and comments taken from the DEM file. Manifold takes the name for the component from the name embedded in the SDTS file. If we double-click open the surface component, what we see in the surface window is determined by the settings in the **View - Display Options** dialog.

The illustration above shows the default **Height** display with **Shading** and **Autocontrast** boxes checked. By default, Manifold will show the surface as a shaded surface as if it were illuminated from the specified direction (an **Azimuth** of 90, East, sunrise, by default). Very cool! This is probably the most frequently used display option for surfaces. In the Northeast quadrant we can see a linear valley cutting across from the northern edge of the surface to just below the middle of the Eastern edge of the surface. That valley marks the path of the San Andreas fault, locus of the famous 1904 earthquake that destroyed San Francisco.
If we choose the **Height** display and uncheck the **Shading** and **Autocontrast** boxes we get a simple grayscale display that shows lower elevations as darker tones and higher elevations as lighter tones.

Choosing **Aspect** for display option we see the surface color-coded to show the aspect of each pixel. **Aspect** is the direction in which terrain faces. Aspect is the direction in which the surface at that point faces. We would use an aspect display to find, for example, all of the southern-facing regions that would provide a sunnier place to locate a house.
Because aspect displays are difficult to interpret if too many aspects are shown at once, they are usually formatted with a simplifying palette. The display above uses the Black to White palette to reduce the number of aspects that are displayed into four major ranges.

Choosing **Slope** shows the inclination at each location, with lighter pixels indicating greater slopes. **Slope** is the angle that the perpendicular to the face makes with the vertical. Black pixels show flat regions. Note the black color of the regions covered by the lakes in the valley of the San Andreas fault. Since lakes are flat, the slope in these regions is flat. **Slope** is a good way of finding reasonably level regions to use for building sites, helicopter pads, weapons emplacement and so forth. It may also be used to find paths through terrain that do not traverse steep ground.
Palettes

Surfaces may be automatically colored by palettes other than the default black and white palette.

Applying the Altitude palette will color the display by elevation to produce the effect seen above. The Shading and Autocontrast options have been checked.

Fixed Interval Palettes

Palettes in Manifold occur in two forms:

- Relative palettes, where colors from the palette will be "stretched" and interpolated as necessary for use in the number of breaks specified in the Colors pane. The example above uses relative palettes.
- Fixed palettes, where colors are associated with specified intervals. When a fixed palette is used in thematic formatting, applying the palette will automatically create as many intervals as are required by the numbers specified for the palette.

Fixed palettes are most frequently used to color surfaces that show terrain elevations. They allow a standard color scheme to be applied for specific elevations that is the same from surface to surface. Manifold includes a few fixed palettes for use with surfaces. Altitudes are predictable since the general elevation of the Earth is known and covers a fairly narrow range. Standardized palettes may therefore be created for altitudes and incorporated into Manifold.

The fixed interval palettes provided in Manifold are named Altitude... and are provided in three variations: a palette where color intervals have round numbers intended to indicate meters, where round numbers are intended to indicate feet and a relative version of the same palette.

Surfaces usually originate in either meter-based data sets, where the elevation values are given in meters, or in foot-based elevation values such as US SDTS DEMs where the elevations are in feet. It is convenient to have the same fixed palette available with both meter-based as well as foot-based palettes so that the color scheme can be used for the same elevation above or below sea level regardless of whether the numbers in the surface represent meters or feet.

It is also convenient to have a version of a fixed palette available as a relative palette. This is allows a quick look at how the full range of palette colors appears in use. We can use whatever data set we want and even if it only ranges between -100 and 1000 we will still see the full set of colors.
The Altitude and Altitude and Bathymetry palettes are typical of those used in large printed atlases. The colors are organized to provide good displays throughout most of the world. Since most world atlases are in meters, the elevations chosen for the color intervals will be round numbers in meters.

The Altitude, Aeronautical palette provides a simple color scheme similar to that used in NOAA sectional aeronautical charts in the United States. This palette has elevations chosen for color intervals that are round numbers in feet.

**Customization**

Palettes used in thematic formats may be customized, and new palettes may be added to Manifold. See the Customization topic.

**Tech Tip**

Change the appearance of surfaces by selecting parts of the surface and then deleting them and letting the background show through. For example, one way to show lakes or other water bodies in surfaces (which are at the same elevation) is to select their pixels and then delete them. Make the background of the surface blue so that the water regions are all colored blue no matter what their elevation.

For example, suppose we start with a surface that's been colored with a palette using View - Display Options. The background for this surface is turned on in the Layers pane. The background has been set to blue color in the View - Properties dialog for this surface.

We click on regions using SHIFT - touch select to select all portions of the surface at a given elevation. Using Add to Selection mode, click on the desired elevation and on regions below the desired elevation.
Press **Delete** and the selected pixels disappear, allowing the blue background to show through. Note how this method provides a "waterline" effect to instantly show which regions are above a given elevation.

This method can also be used to remove flatter parts of a surface, leaving only very mountainous areas. The surface can then be used in a map overlaid upon drawings to provide visual cues of mountainous terrain.

### A Note on Computation of Slopes

There are many ways to compute the "slope" of a surface. If a surface consists of an even, flat plane tilted in only one direction that extends over a great distance then most people and algorithms would agree what the "slope" is. However, in reality surfaces often consist of numerous undulations in small scale and large and the "slope" computed depends upon how the computation is made. "Slope" is an artificial approximation based upon a subjective judgement or other heuristic as to what plane should be taken as an approximation for a particular surface or a portion thereof.

At the lowest level, surfaces are ordered sets of pixels arranged in a regular array with a height for each pixel. To compute the slope of a surface one must consider some window, that is, some subset matrix of pixel values, and then perform a computation on it. Let's consider three possible ways to compute the slope of a surface given a $3 \times 3$ pixel window.

Consider the following window:

\[
\begin{array}{ccc}
  z_1 & z_2 & z_3 \\
  z_4 & z_5 & z_6 \\
  z_7 & z_8 & z_9 \\
\end{array}
\]

To compute the slope we have to approximate this window with a plane that will give us $dx$ and $dy$ factors. We can then use the $dx$ and $dy$ factors to compute the slope (a simple matter of $\tan(\text{hypot}(dx, dy) / 2)$ with some scale correction).

Here is one way to do this:

\[
\begin{align*}
  dx &= z_6 - z_4 \\
  dy &= z_2 - z_8
\end{align*}
\]

Here is another way to do this:

\[
\begin{align*}
  dx &= ((z_3 - z_1) + (z_6 - z_4) + (z_9 - z_7)) / 3 \\
  dy &= ((z_1 - z_7) + (z_2 - z_8) + (z_3 - z_9)) / 3
\end{align*}
\]

Here is yet another way to do this ($k > 1$):

\[
\begin{align*}
  dx &= ((z_3 - z_1) + k*(z_6 - z_4) + (z_9 - z_7)) / (2 + k) \\
  dy &= ((z_1 - z_7) + k*(z_2 - z_8) + (z_3 - z_9)) / (2 + k)
\end{align*}
\]

For example, in the above case one might use $k = 2$ to weight the central comparison more. There are numerous other ways as well, but the above three show in simple fashion three possible approaches.

If different software uses different methods, it will provide different values for "slope" at any given location. In general, Manifold uses the second method but there are software packages that use the first method, the third method and many other methods.

### Example: Transferring Aspect to Points

Manifold's raster extensions to SQL allow us to exploit computations on surfaces from within SQL. For example, suppose we have a drawing of points that are located in the same region of interest as a surface and we would like to compute the aspect of the surface at each point and to save that value as a data attribute for each point. This is easy to do using SQL as follows:
1. Open the drawing's table and add a new floating-point column named Aspect.
2. Create a new query.
3. Open the query and enter the following text for the query (substitute the name of the drawing for "Drawing" and the name of the surface for "Surface"):  
   \[ \text{UPDATE } [\text{Drawing}] \text{ SET } [\text{Aspect}] = \text{AspectHeight}([\text{Surface}], \text{Centroid}([\text{Geom (I)}])); \]
4. Run the query. This will set the values in the Aspect column for each point to the aspect of the surface at that point's location.

**See Also**

Using Surfaces as Images - Surfaces are virtual views either in 2D or 3D. They can be dragged and dropped into maps as is. They can also be copied and pasted as images to allow subsequent effects using image manipulation tools.

Stylistically, the Display Options dialog has much in common with the thematic formatting dialog. See Thematic Formatting for examples.

See the Displaying Data in a Gradient Map topic for an example that shows demographic data displayed in a continuously varying gradient throughout the United States using a surface.
Using Surfaces as Images

When a surface is opened in a surface window, Manifold computes the view to present based on the settings in View - Display Options for that surface. The views presented in the surface window are the results of computations and usually represent greater information than simply color coding the values in the Surface. To capture the view presented, we can Copy the surface and Paste As an image. We can then use the resulting image as desired.

To Capture a Surface as an Image

1. Open the surface in a surface window by double-clicking on the surface in the project pane.
2. Use View - Display Options to specify the appearance of the surface.
3. At any time thereafter, Copy the surface component in the project pane and Paste As an image. The resulting image will capture the appearance of the surface component using the current display options.

Example

See the Surface Display Options topic for an example of importing the Montara Mountain USGS high resolution SDTS - DEM as a surface.

Importing the file creates a surface component and a terrain, as well as some comments components.

We can double-click open the surface and set the View - Display Options to show Height with Shading and Autocontrast as shown in the Surface Display Options topic.

In the project pane we can right click on the surface and choose Copy. We can then right click into the project pane and choose Paste As - Image.

A new image appears in the projection pane. It is a grayscale image capturing the view of the surface seen in the surface window.
If we open the resulting image we can see it is the same as the surface window view. However, we now have a real image. We can paint into it, we can covert it into an RGB or RGBA image and we can use it as a layer within a map. We could use the original surface as a layer in a map too, but converting it into an image allows us a wider range of editing effects such as partial transparency applied to individual pixels as can be done with RGBA images.

For example, we could create an image from a surface and then use it within a map as seen below. The map was created using DTED Level 2 and VMAP Level 2 sample data downloaded from the NIMA web site.

Create the image above using the following steps:

1. Import a surface from DTED Level 2 sample file for Kileen, Texas.
2. Import the VMAP Level 2 file for the same area, checking hydrography and roads layers.
3. Open the surface and set display options to use Height, Shading and Autocontrast.
4. Copy the surface and paste as an image.
5. The VMAP Level 2 importer creates hydro and roads maps. Open the hydro map.
6. Drag and drop the image into the hydro map. Place it at the bottom of the map layer stack.
7. Drag and drop the roads layer into the hydro map.
8. Format the water areas in the hydro map in blue colors. Set their opacity to 50%.
9. Format labels and hydro lines to taste.

Note that importing a surface usually results in a georeferenced surface, since almost all formats used for surfaces are geographically aware formats. When a georeferenced surface is copied and pasted as an image, the resulting image is also georeferenced. It may be immediately combined with drawing layers in maps without further georeferencing.
Linked Surfaces

Linked surfaces are surfaces that are dynamically created from tables or queries. The tables and queries used to create linked surfaces may be inside the Manifold project or they may be in external databases. Linked surfaces are shown in the project pane using an icon that includes a yellow "database" cylinder to show they are created from a database table or query. Linked images are normally read-only.

Linked surfaces may also be created in a two step process, where the virtual table for an existing surface is used in a query to manipulate that surface or to fetch part of the surface based upon desired criteria. A linked surface can then be created from that query. This technique is often used within Manifold projects that will be used in an IMS website.

Linked surfaces may also be created from tables or queries stored in external databases. For example, we could use a query to select all columns for all pixels in an surface's virtual table and then export that query as a table into an .mdb file or to some other database storage.

When a surface is linked from a table or query, each record in the table or query produces a pixel. The coordinates of the pixel are taken from the X and Y columns (cast to integer) specified in the Link dialog. The height of the pixel is taken from the Height column.

Editing Linked Surfaces

Although a linked surface that appears in the project pane is normally read-only, it is usually possible to edit that linked surface by editing the data source from which it is created. For example, suppose we have a linked surface created from a query that uses the virtual table of another surface. We could change that linked surface by editing either the original surface or by editing the query.

Relink and Unlink

If a connection is lost between a linked surface and its originating data, the Surface - Relink command allows us to restore the connection. If we would like to convert a linked surface into a regular surface, the Surface - Unlink command will sever all connections to the originating table or query and will convert it into a regular surface.

See Also

Virtual Tables for Images and Surfaces
Queries and Images or Surfaces
Raster Extensions
Linking Surfaces
Terrains

Terrains are 3D views of Surfaces. Surfaces are raster data sets that may be seen either in 2D or in 3D. The default view for a surface is as a 2D image. Clicking open the surface will show the surface in a 2D view like an image.

When a surface is imported, a terrain component for that surface is automatically created.

Terrains are shown under their parent surface in the project pane hierarchy. Creating another terrain for the same surface will also show it under the parent surface.

Create a terrain for a surface by right clicking the surface and choosing Create - Terrain.

Additional terrains may be created for a surface and each such terrain can have its own coloring and other characteristics. Creating an additional terrain for a surface does not require any more storage space since terrains are just different views into the one data set contained in the surface. An unlimited number of terrains can be created for each surface. If desired, the default terrain created for a surface may be removed so that the surface has no terrains.

Clicking open a terrain will present a 3D view of the data in the surface.

Once a surface is imported into Manifold, we can see it as an image by opening the surface, or we can see it in a 3D terrain window by clicking open the surface's terrain in a terrain window.

When open, the terrain window shows a view of the terrain from a particular viewpoint, as if seen from a camera at that position. The initial view is seen from a camera position at the center of the data set. What is seen in the window may be adjusted using keyboard navigation commands to alter the position of the camera, the direction it is pointed, and the field of view. In addition, we can use Linked Views to navigate to different locations in the terrain.
Controlling the Appearance of the Terrain

The contents of the terrain window are controlled by menu options that appear under the Terrain menu. These options also appear in the context menu when right clicking into a terrain window.

- **Open Surface**
  - Open the parent surface for this terrain.

- **Surface**
  - Specifies how the data surface that defines the terrain should be visualized. Includes options such as the texture used for the surface, the color determined by the elevation, the level of details computed, the size of the data set in view and whether the surface is shown with walls and in wireframe style or not.

- **Overlay**
  - Overlay the terrain surface with images or with drawings. Specifies options to be used to display objects from drawings.

- **Clouds**
  - Display synthetic "clouds" in the sky in accordance with given options.

- **Fog**
  - Apply a haze effect that can appear as fog or as a subtle haze in the distance. Provides a greater sense of reality and depth.

- **Lighting**
  - Choose options for creating highlights and lowlights in the surface to represent the angle of lighting.

- **Water**
  - Create an opaque or semi-transparent "waterline" plane of given color. Used to simulate waterline effects.

- **Snap to Surface**
  - Forces the "camera" from which the terrain window is seen to hover just above the surface by applying slight gravitation to camera movement.

**Note:** The Snap to Surface function assumes the camera begins above the surface so that gravity pulls it down to the surface. If the initial camera position is below the surface the camera must first be moved above the surface using a keyboard Q command.

**North Arrow**

In addition to the above controls, use View - North Arrow to show or hide the "compass" North arrow in the terrain window. We can also right click on the North arrow in the terrain window to hide it or to change alignment or properties.

**Saving Terrain Views as Images**

The Tools - Make Image command captures the current scene displayed in the terrain window as an image. This command allows us to specify the pixel dimensions, and thus the detail seen, of the image.
For example, suppose we open a terrain window like the one above. This shows a view into a terrain created from the Montara Mountain sample surface. We can use Make Image to create an image of this view, for example, if we wanted an image to use in a brochure or to include in some other document.

Suppose, however, we would like to capture this view at higher resolution that seen at screen resolution on our computer monitor. If we save the image at screen resolution and then we print an enlarged version of the image within a document such as a poster, the image would appear overly pixilated. To see this effect, consider the region of the view marked by the red box.

When enlarged (as might occur when printing the image in a large size in a poster) we can see the detail in the terrain is limited by the screen resolution of the view. We can create the image with resolution higher than screen resolution by simply specifying a larger number of pixels to use for the image in the Make Image dialog.
For example, we might specify that the image be created in a pixel size of 2325 by 1450 pixels, about five times the resolution of the original terrain view at screen resolution.

In that case, when Manifold creates the image from the terrain view it internally will re-compute and re-render the terrain view at higher resolution so that the view will be seen in the image at the desired number of pixels horizontally and vertically. This shows the same view with greater detail.

The view above shows a small portion of the resultant image (which is too large to appear in Help), zoomed so that it shows approximately the region marked by the red box in the original terrain view. We can see that the distant mountain peaks that were rendered in an overly pixilated manner at screen resolution are now rendered in much higher detail.

**Saving Terrain Views in the Views Pane**

Terrain views may be saved in the Views pane to go back to the same view at a future time.

**Keyboard Navigation**

In addition to the keyboard arrows, Manifold uses keyboard navigation shortcuts to control the view seen in the Terrain window.
Surfaces and Terrains

W  Move forward.
S  Move backward.
Q  Strafe Up. (Move vertically up)
E  Strafe Down. (Move vertically down)
A  Strafe Left. (Move horizontally left)
D  Strafe Right. (Move horizontally right)

SHIFT  Pressing the SHIFT key with any of the above key commands will increase the effect of that command.

Up / Down Arrow  Tilt view up / down to +60 or -60 degrees from horizontal.
Left / Right Arrow  Rotate view left / right.
+  Increase field of view (up to 130 degrees).
-  Decrease field of view (down to 30 degrees).
*  Reset field of view to 90 degrees
/  Move camera position to center of terrain.

The keyboard shortcuts are designed to allow "two handed" fast keyboard navigation as favored by experts in various computer tactical games. Strafe is gamer-speak for moving sideways or up/down without rotation. A Q move is thus the same as moving straight up, as in a helicopter.

To fly upward from the terrain surface, make sure that Terrain - Snap to Surface is unchecked. Next, either use Q to fly directly upwards or use the keyboard up arrow to rotate the view upwards and then use W to move forward. The Q move lifts us up as if in an elevator or helicopter moving straight up. The up arrow followed by a W is like angling an airplane upwards and then flying forward and up the angle.

Keyboard Shortcuts for Z Scale in Terrain Windows

With the focus on the terrain window we can use the following keyboard shortcuts to modify Z scale.

Page Up  Increase Z scale by .01
SHIFT-Page Up  Increase Z scale by .10
Home  Set Z scale to 1.0
Page Down  Decrease Z scale by .01
SHIFT-Page Down  Decrease Z scale by .10

Performance Notes

Manifold's terrain view window uses the OpenGL subsystem in Windows for 3D visualization. Manifold requires a functioning OpenGL subsystem to display terrains. If there are no OpenGL capabilities in the system terrain windows will be blank when opened.

The visible performance of a terrain window is determined by two factors. The first factor is the time required for Manifold to access data and to generate parameters for rendering (heights, colors and textures). The second factor is the time required by the OpenGL system to process the generated data. Because Manifold is so fast at computing terrains, quite frequently the limiting factor in processing speed is the speed of the OpenGL system, not the time it takes Manifold to fetch data and set up the view.

It is therefore critically important that your OpenGL system is as fast as possible. The speed of the OpenGL system will depend upon the ability of the video graphics card to support OpenGL in hardware as well as upon the
quality of the OpenGL drivers in use, which are normally provided by the graphics card vendor. For terrains, charts and other 3D work, get a fast graphics card with OpenGL support in hardware and lots of local graphics memory. The manifold.net team recommends cards based on NVIDIA chips. It is critically important to use good drivers. Efficient, well-written drivers are more important than raw hardware capability.

Check your graphics card vendor's web site for their latest drivers for your version of Windows. Check also the web site for the chip vendor who makes the graphics chip used by your graphics card. The chip vendor will often provide faster drivers than those available from the card vendor. Using good drivers can make the difference between smooth "fly through" motion and very slow, jerky motion in terrain views.

For example, during the development of Manifold System a number of low-cost (under $100) AGP graphics cards using NVIDIA chips were installed to test terrain window performance. When using the drivers packaged with the cards the terrain window moved painfully slowly. Several seconds were required for each slight scene change when a key was pressed to move forward, backward or to rotate left or right. After downloading and installing the latest graphics driver from the NVIDIA web site the terrain window provided smooth motion, "fly-through" quality graphics. The graphics motion seemed literally hundreds of times faster with the latest driver.

Because the quality of the graphics driver has such a great effect on the performance of the card it can be difficult to purchase a graphics card on the basis of hardware specifications only. Some of the best-known brands in graphics cards have surprisingly poor graphics drivers for OpenGL and DirectX, and some of these best known brands are also very poor at providing timely support for new versions of Windows such as Windows XP x64.

An especially frustrating situation can arise when a chip vendor sells a special version of a common graphics chip to a laptop vendor for use in a built-in graphics capability. In such cases it is frequently the case that the drivers initially shipped with the laptop are not as fast as those developed later on for the graphics chip. It is possible that users will have to wait for the chip vendor to release a special version of updated drivers to the laptop maker and then wait for the laptop maker to release the updated drivers to the user. In some cases, even famous name laptop makers have "orphaned" some products by failing to provide improved drivers for their graphics systems or even by failing to support more recent versions of Windows, such as XP.

One strategy for avoiding such problems is to buy laptops that use NVIDIA chips, because NVIDIA has in the past taken responsibility for the good performance of its products no matter how a subsequent manufacturer has used them, and has posted updated drivers on the NVIDIA website that can be used for all of its chips, even in laptops. In particular, NVIDIA has made it very easy to get 64 bit drivers, including 64 bit drivers that work correctly with AMD dual core 64 bit processors.

If you are assembling a new system, as graphics cards have increased in speed it has become difficult to buy a new graphics card that does not display Manifold terrains with great speed. If you use a reasonably recent (no need to buy the latest, hottest version) NVIDIA based card from a vendor who advertises support for new versions of Windows and OpenGL and if you also utilize the latest NVIDIA drivers, the rendering speed of terrains will be very good.

For maximum terrain viewing performance, use SLI-capable NVIDIA PCI Express graphics cards in an SLI-capable motherboard to team up two graphics processors for rendering. Prices on graphics cards are dropping rapidly: as of this writing, installing two high-end, SLI-capable NVIDIA graphics cards with 256 MB of RAM each costs a total of $250, an amazing deal for the resultant throughput. It is often faster to use two cards via SLI than it is to spend disproportionately more money for a single card that uses the very latest, super-hot graphics chip. For example, two SLI cards using slightly downrev, but still awesome chips might be had for $150 each and the combination could end up being as fast as or faster than the latest superchip board at $750 each. Of course, if money is no object, get two of the latest boards!

A typical machine at manifold.net used to explore large terrains is an AMD Athlon 64 x2 with four gigabytes of RAM running Windows XP x64 or Windows Server 2003 x64. It will have a motherboard that can host two SLI cards with 16x PCI Express slots and run two graphics boards using whatever is a notch or two down from NVIDIA’s latest chipset. Such machines can twirl full-sized terrains around effortlessly with perfectly smooth motion and no delays.

See the discussion in Performance Tips and in Using RAM and other Machine Resources for more on optimizing performance.

**Tech Note**

The speed with which Manifold sets up the terrain rendering job depends on the dimensions of the underlying surface, so larger surfaces will generally be slower than smaller surfaces. However, the speed penalty is semi-logarithmic: the time to generate the rendering data for a surface P that is two times as large as a surface Q is much smaller than two times the time required for surface Q. Therefore, although the speed of terrains does
depend somewhat upon the size of the underlying surface the dependency is non-linear so that greatly scaling up
the size of a terrain does not have that great an impact on rendering speed. The greatest impact will be the
speed of the graphics system.

See Also

Linked Views - Most navigation in terrains will be accomplished using the surface window as a "mini-map" via the
linked views capability of Manifold. Please take a moment to read the Linked Views topic. No matter how expert
one becomes in using keyboard shortcuts, usually using linked views is by far the fastest and easiest way to
navigate the view in a terrain window.

Terrain - Surface
Terrain - Overlay
Terrain - Clouds
Terrain - Fog
Terrain - Lighting
Terrain - Water
Overlays

Terrains may be overlaid with images, drawings, labels or even other surfaces using the Terrain - Overlay command. See the Terrain - Overlay topic for details on controls in this dialog.

To overlay a component onto a terrain:

1. Georegister any components to be used so they are in the same geographic location as the surface and terrain being used.
2. Open the terrain in a terrain window.
3. Choose Terrains - Overlay from the main menu.
4. The upper pane shows components available in the project for overlay. Check those components to be overlaid.
5. Choose display options for showing points, lines and areas.
6. Press OK.

Overlaying several components or complex components can require much computing time and will result in a long delay (potentially many minutes on a slow machine) while the overlays are rendered in the terrain window.

Example: Overlay a drawing

Using File - Import - Drawing with SDTS Files type we've imported the 1:24K scale SDTS DEM for the Montara Mountain, California, USGS quad. We've also imported the roads lines from the equivalent 1:24K scale SDTS DLG for this same quad. See the Combine a Surface and a Drawing in a Map example topic for details of import.

The drawing is a typical vector drawing. We've imported only the lines from the SDTS file.
We can open the surface and use View - Display Options to color it with a palette. The illustration shows a view zoomed into the upper right corner of the surface onto San Andreas lake in the valley of the San Andreas Fault.

We can double click open the terrain, color it using Terrain - Surface and then navigate to a position showing a view to the northwest down the axis of the San Andreas Fault.
To overlay the terrain with objects from the drawing we choose Terrain - Overlay. In the Overlay dialog we check the box for the roads drawing and check the Paint over terrain box as well as the Smooth line body box (for pretty, anti-aliased lines). We uncheck the Points boxes since there are no points in the drawing we will overlay. Press OK.

The result shows lines from the drawing overlaid on top of the terrain. If the drawing had any points we could have checked the Paint over terrain option for points and those too would have appeared.

**Example: Overlay a surface**

At first glance it may seem strange to overlay a terrain with a surface. After all, aren't terrains a 3D view of a surface? It makes sense to overlay a terrain with a surface because this allows creating a terrain with one surface while coloring it with data from a different surface.

Suppose for example we have a DEM that we use to create a surface and thus a terrain that shows elevation. Suppose we also have another surface that shows abstract data, such as infrared reflectivity, temperature, samples of pollutants, or some other data that we have colored as an abstract data surface. We can open the terrain to see the 3D lay of the land and we can also color the surface by overlaying a surface onto that terrain.
Using the Montara Mountain DEM again we've imported a surface and have opened the terrain in a window. The illustration above shows the terrain with no palette or texture applied in a view looking down the San Andreas Fault.

Suppose we also have another surface in our project that gives temperatures as measured by airborne sensors during an overflight. We see that surface opened above colored with a palette and using shading and autocontrast to give it apparent depth, as if it showed terrain elevation.
To color the terrain using the temperature scan surface, we open the **Terrain - Overlays** dialog and check that surface. The **Areas**, **Lines** and **Points** options are irrelevant when only images or surfaces are overlaid upon the terrain, so it doesn't matter what we use for them.

The result is that the terrain is colored by the surface overlaid upon it. We now see not only the terrain elevations but also their temperatures.

**Example: Displaying Contour Lines on a Terrain**

Importing the **hypsoigraphy** modules from SDTS files will result in drawings showing detailed contour lines. These may be overlaid onto terrains as well.
The illustration above shows a terrain with no overlays and no palette or texture. The illustration below shows the same terrain with contour lines from a 1:24K-scale SDTS DLG hypsography module. High resolution detailed drawings such as 1:24K-scale SDTS DLG hypsography files can take a long time to render. This example uses the Montara Mountain SDTS files and took 10 minutes on a 600Mhz PIII machine to overlay onto a terrain.

Manifold, of course, can also produce Contour areas that can be overlaid onto terrains.

**Notes**

Manifold uses OpenGL to render 3D views within terrain windows. A fast graphics card with well-implemented OpenGL drivers using hardware acceleration is a must.
On occasion one encounters graphics cards with bugs in their OpenGL drivers. The appearance of fine grid lines like those above when a terrain is overlaid with an image or a surface is a typical bug. The previous images were rendered on an NVIDIA card while the screenshot above was rendered using an ATI card. To remedy such problems, try downloading the latest drivers for your graphics card from either the card manufacturer or the manufacturer of the graphics chip used on the card.

If new drivers don't solve the problem consider switching to a different graphics card. The manifold.net team recommends cards with NVIDIA graphics chips since NVIDIA at the present writing has consistently turned out the best drivers for OpenGL and other graphics standards.

The "temperature scan" surface is fake data contrived for the purpose of this example. If it were real, the body of the lake would be cooler than the land.

**Tech Tip**

In drawings and labels, setting the foreground color to be transparent color will hide the entire object or label. Terrains respect this convention: if a drawing object or label has foreground color set to transparent color it will not appear in a terrain overlay. Likewise, if an area border foreground color is set to transparent color the area walls will also be hidden.

**See Also**

Area Overlays - Areas overlaid on terrains with 3D walls.
Terrain - Overlay - For examples of overlays of labels and surfaces and other detailed examples.
Area Overlays

When drawings that contain areas are overlaid upon terrains, Manifold will "float" the areas above the terrain and, if desired, render walls for the areas to give the appearance of extruded shapes. Areas take their color from the area background color and walls take their color from the border foreground color specified for the overlaying drawing. When representing areas as overlays, area style and foreground color are not used.

Such overlays are particularly handy for the schematic representation of buildings and other structures in 3D views of terrains.

Area overlays are turned on using the controls in the Terrains - Overlays dialog when a terrain is open.

Example

Consider a map with a drawings layer that appears above a surface. The surface is the Montara Mountain sample surface that has been colored using the View - Display Options dialog using the Altitude and Bathymetry palette.

<table>
<thead>
<tr>
<th>ID</th>
<th>Height</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>50</td>
</tr>
<tr>
<td>2</td>
<td>30</td>
</tr>
<tr>
<td>3</td>
<td>50</td>
</tr>
<tr>
<td>4</td>
<td>60</td>
</tr>
<tr>
<td>5</td>
<td>50</td>
</tr>
<tr>
<td>6</td>
<td>75</td>
</tr>
<tr>
<td>7</td>
<td>60</td>
</tr>
<tr>
<td>8</td>
<td>80</td>
</tr>
<tr>
<td>9</td>
<td>40</td>
</tr>
<tr>
<td>10</td>
<td>40</td>
</tr>
</tbody>
</table>
The drawing consists of areas only, and has only one column in the drawing's table, a **Height** column that gives the height of each area above the local ground level. Note that this value is not an absolute height above sea level, but rather is intended to represent a height above ground.

![Format](image)

The drawing layer has been thematically formatted for area background color using the **Height** value so that each area is colored according to the height. The area border has also been colored by the same thematic format.

![Overlay](image)

If we open up the terrain for the surface and choose **Terrain - Overlay** we can set up the **Overlay** dialog to show area overlays. We click on the **Paint over terrain** box and make sure the **Walls** and **Vary height by column** boxes are also checked. The latter box will vary the height at which areas are rendered over the terrain by the contents of a column named **Height**.
The result is the scene illustrated at the beginning of this topic. Areas have been rendered using the colors specified in the thematic format for area background color for each area. Walls have been rendered using the area border foreground color. Area walls and areas are rendered using the Terrain - Overlay dialog settings for Opacity for areas (100% in this case), as well as lighting specified for the terrain.

If we were to change the Opacity setting for areas in the Terrain - Overlay dialog to 50% we could see that the extruded 3D area shapes shown in the terrain become partially transparent.

Area overlays may be combined with other effects, such as overlay of images. The above scene was created by first overlaying a TerraServer image upon the terrain and then overlaying a drawing containing areas showing building outlines. The area objects have a Height field giving the height of each building. (This image was created at Cornell University and is used by permission.)

**Tech Tip**

In drawings and labels, setting the foreground color to be transparent color will hide the entire object or label. Terrains respect this convention: if a drawing object or label has foreground color set to transparent color it will not appear in a terrain overlay. Likewise, if an area border foreground color is set to transparent color the area walls will also be hidden.

**See Also**

Overlays
Terrain - Overlay
Contours

The Contours dialog works with surfaces to create a drawing that includes areas representing contours from the surface.

To create contours:

1. Open a surface in a surface window, or click on a surface layer in a map.
2. Choose Surface - Contours from the main menu.
3. Supply a Name for the drawing to be created and (optionally) a description.
4. Choose the type of contours to be created in the Create box.
5. Specify the contour values to be used in the Height pane.
6. Press OK.

The Add button is used to add one new contour interval.

To specify values to be used in the Height pane:

1. Click the Add button.
2. Double click into the resulting new row and change the Height value to the value desired.
3. Organize the heights by using the Move Up / Down / to Top / to Bottom buttons.

The Add Sequence button automatically creates a series of contour intervals.

To specify a series of contours automatically:

1. Click the Add Sequence button.
2. In the Add Sequence dialog, specify the Step between intervals and press OK.
3. A series of rows with heights from the lowest to the highest at the specified interval step will be created automatically.

Note that the Contour dialog refers to "heights" because most often in a GIS context it is used with surfaces that represent terrain elevation data. It can also be used to create contours with abstract data where the value in the surface represents some other value such as temperature or population.

<table>
<thead>
<tr>
<th>Name</th>
<th>Name for the drawing to be created that will contain the contours.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Optional description for the drawing.</td>
</tr>
</tbody>
</table>
Surfaces and Terrains

Create
How contours will be created:

Areas above heights - Create contour areas that show all locations higher than a particular contour height. Contour areas for higher heights will overlap contour areas for lower heights. This is the “wedding cake” effect.

Areas below heights - Create contour areas that show all locations lower than a particular contour height. Contour areas for lower heights will overlap contour areas for higher heights. This is the “inverse wedding cake” effect.

Areas between heights - Create contour areas that show all locations with height between the height intervals given. No contour areas will overlap. The most frequently used option when creating contour areas.

Lines - Create lines at a given contour level. The classic contour line effect use within printed topographic maps.

Heights Pane
A list of heights to be used for contours. Can also be used to specify abstract values when a surface represents data other than terrain elevation.

Add - Add a new height range from the contours to be created.

Add Sequence - Automatically create a series of contour intervals.

Delete - Delete highlighted height range from the contours to be created.

Move to Top - Move the highlighted height range to the top of the heights pane.

Move Up - Move the highlighted height range up one position in the heights pane.

Move Down - Move the highlighted height range down one position in the heights pane.

Move to Bottom - Move the highlighted height range to the bottom of the heights pane.

(status info) The bottom of the dialog will report the minimum and maximum heights found in the surface.

Example

Visiting the USGS web site we’ve downloaded the Montara Mountain 1:24K-scale SDTS file that provides a DEM module for terrain elevation. We’ve imported it as a surface. Open the surface in a window.
Choose **Surface - Contours** to launch the **Contours** dialog.

In the **Contours** dialog click **Add** to add a new row in the **Heights** pane.

Double click into the new row that’s created and add a **Height** of 3.

Click **Add** again and add a **Height** of 100.

Click **Add** and add **Heights** at levels of 250 and 500 as well. Press **OK**.
The result is a new drawing in the project. If we open the drawing we see it consists of non-overlapping areas, where each area shows the contours between a given height range. The drawing's table has one field, the **Height** field, which gives the value of Height used to create each area.

We can thematically color the new drawing using thematic formatting controlled by the **Height** field, or simply by using the Color dialog to quickly color the areas with different colors.

For a nice effect, we can create a map using the original surface and then drag and drop the thematically colored contour drawing into the map in a layer above the surface. We then change the opacity of the drawing's layer to 50%. This creates a visual mixture of the thematically formatted contours and the surface shading.
Another classic map can be created by choosing the Lines option in the Create box in the Contours dialog and to then layer the resultant contour lines in a map above the surface, or to use contour lines drawing in a map above a different drawing.

Using the Add Sequence Command

The Add Sequence button allows us to add a series of contour heights.

Suppose we would like to add many contours at once for a surface that has a minimum height of 389 feet and a maximum height of 1708 feet. We launch the Contours dialog and press the Add Sequence button.
In this example we will start the series at 1000 feet by entering that value in the **Offset** box and we will enter **100** in the **Step** box to create a contour every 100 feet.

![Contours dialog box]

The result is a sequence of heights in the **Heights** pane that are 100 feet apart beginning at the 1000 foot level.

**See Also**

**Surfaces and Terrains**
Editing Surfaces

Surfaces may be edited using many of the same methods used to edit images. Methods include:

- Changing the surface "free hand" by making selections and the deleting pixels or otherwise modifying them.
- Using the Transform toolbar to change many pixels at once throughout the surface.
- Using effects or adjust commands from the Surface menu such as Quantize to edit the surface.
- Changing the surface by projecting or georeferencing it.
- Altering surfaces with scripts or other tools.
- Use View - Display Options to change the appearance of the surface, including computing slope and aspect.
- Creating new surfaces from drawings or other components via copy and paste as.
- Cutting, copying and pasting between surfaces.
- Copying a surface, pasting as a table, changing the table values and then creating a surface from the table.
- Edit the image using update queries that operate on the image's virtual table to directly change the colors of pixels or other characteristics. See the Virtual Tables for Images and Surfaces topic as well as the Queries and Images or Surfaces topic.
- Use the Surface - Transform command dialog packaged with the optional Surface Tools extension to perform arbitrary transformation of surfaces, including computations that involve multiple surfaces such as subtracting one surface from another.
- Use commands on the Surface Menu to change the surface.

Cropping a Surface

Cropping a surface to a given size by making a selection and then choosing [Selection] Crop in the transform toolbar and pressing Apply. The surface will be cropped down to the minimum enclosing rectangle of the selection.

Use Crop Margin to crop the surface by the given number of pixels at the border. Use Add Margin to expand the size of the surface by the given number of pixels at the border.

Eliminating "No Data" or spurious values

Some surfaces may contain pixels coded with a value of -9999 or some other distinctive value to indicate there is no reliable data for that pixel. To eliminate such pixels we have two options:

- Select pixels representing no data using Select Touch or other selection method and delete them.
- Use the Surface - Transform dialog provided in Surface Tools to cut out values lower or higher than the given threshold.

Copying a Surface and Pasting as a Table

Surfaces may be copied and pasted as tables. When pasting a surface as a table, the system will compute intrinsic fields for each pixel in the surface. The Paste As dialog offers the following choices:

- Select All - Paste all fields.
- Select None - Do not paste any fields.
- Select Inverse - Do not paste all current Paste As fields and paste all other fields. A fast way to use all but one field: click Select None, specify a Paste As choice for the one field not desired and then click Select Inverse.

Center Easting (I)  X coordinate of the center of this pixel in native coordinate system units adjusted with the values of the local scale and local offset parameters.
**Center Latitude (I)** Latitude of the center of this pixel in degrees latitude.

**Center Longitude (I)** Longitude of the center of this pixel in degrees longitude.

**Center Northing (I)** Y coordinate of the center of this pixel in native coordinate system units adjusted with the values of the local scale and local offset parameters.

**Center X (I)** X coordinate of the center of this pixel in native coordinate system units.

**Center Y (I)** Y coordinate of the center of this pixel in native coordinate system units.

**Height (I)** Height value.

**Invisible (I)** 0 or 1 to indicate invisible pixels, normally used for missing height values.

**Latitude (I)** Latitude of the lower left corner of this pixel in degrees latitude.

**Longitude (I)** Longitude of the lower left corner of this pixel in degrees longitude.

**Easting (I)** X coordinate of the lower left corner of this pixel in native coordinate system units adjusted with the values of the local scale and local offset parameters.

**Northing (I)** Y coordinate of the lower left corner of this pixel in native coordinate system units adjusted with the values of the local scale and local offset parameters.

**Selection (I)** Boolean: currently selected or not.

**Selection Mask (I)** A byte giving the saved selections mask.

**X (I)** X coordinate of the lower left corner of this pixel in native coordinate system units.

**X Offset (I)** X position in pixel coordinates from the lower left corner.

**Y (I)** Y coordinate of the lower left corner of this pixel in native coordinate system units.

**Y Offset (I)** Y position in pixel coordinates from the lower left corner.

Copying a surface and pasting as a table is a good way to make sophisticated edits or to perform analyses. For example, one can copy a surface and paste it as a table and then use the transform toolbar to change the values in the Height field. After making modifications, one can then copy the table and paste it back as a surface.

Suppose we would like to know the average height in a surface in regions that are within 100 meters of certain roads in a drawing. We create a buffer zone about the roads in the drawing, select the buffer zone and then transfer the selection to the surface. We can then copy the surface and paste as a table including the Height (I) and Selection (I) fields. We select all records in the table where Selection (I) is 1 and then use a ViewBot to report the average value of Height (I) for all records in the selection.

**Note:** When copying surfaces and pasting as tables, don't forget to paste the Longitude (I) and Latitude (I) fields or the X (I) or Y (I) fields to allow subsequent conversion back to a surface, if desired.

**Tech Tip**

Change the appearance of surfaces by selecting parts of the surface and then deleting them and letting the background show through. For example, one way to show lakes or other water bodies in surfaces (which are at the same elevation) is to select their pixels and then delete them. Make the background of the surface blue so that the water regions are all colored blue no matter what their elevation.
For example, suppose we start with a surface that's been colored with a palette using View - Display Options. The background for this surface is turned on in the Layers pane. The background has been set to blue color in the View - Properties dialog for this surface.

We click on regions using SHIFT - touch select to select all portions of the surface at a given elevation. Using Add to Selection mode, click on the desired elevation and on regions below the desired elevation.

Press Delete and the selected pixels disappear, allowing the blue background to show through. Note how this method provides a "waterline" effect to instantly show which regions are above a given elevation.

This method can also be used to remove flatter parts of a surface, leaving only very mountainous areas. The surface can then be used in a map overlaid upon drawings to provide visual cues of mountainous terrain.

Converting Data Types in Surfaces

Surfaces internally consist of a regular array of numbers. The numbers within surfaces can be any one of a variety of numeric types, either integers of various types (signed, unsigned, 8-bit, 16-bit, 32-bit or 64-bit) or single or double precision floating-point numbers.

The Surface - Convert To command allows conversion between different types of numbers within the surface.

Masks and Surfaces

Masks can be used with surfaces as they are with images. Saving a mask from a surface results in an image that is georegistered to the surface and that retains any selection that was made in the surface. This is a handy way of saving complex selections made in a surface as an alternative to using the Selections pane.

See Also
Editing a Surface for Visual Effect
Creating Surfaces from Drawings and Tables

Surfaces may be created from drawings or tables by copying the drawing or table and pasting it as a surface.

When imported from a surface, image or drawing (for those drawing formats like SDTS that can contain surfaces), surfaces are already created as raster, grid data. At times, however, we might want to create a surface using an irregular pattern of points or from a geocoded table that contains a spatially irregular set of records. To do so we simply copy the drawing or the geocoded table and then we paste it as a surface. We may even copy an image and paste it as a surface if it makes sense to do so. The Paste As Surface dialog will be raised to allow us to specify interpolation method and other parameters of interest in creating the surface.

Pasting Images as Surfaces

Height Specify the values to use for elevation values for the surface.

Type Choose the data type to use for elevation values.

Pasting Tables as Surfaces

X / long Choose which field in the table should be used for an X or longitude coordinate.

Y / lat Choose which field in the table should be used for a Y or latitude coordinate.

Latitude / longitude coordinates Check if the values in the X / long and Y / lat boxes are in degrees latitude and longitude. If this box is not checked the surface will be created in Orthographic projection treating the X and Y values as meter-based coordinates.

Height Choose the field to be used for elevation values for the surface.

Type The data type to use for elevation values in the surface. If the originating field is a different type, Manifold will convert to the given type when writing to the surface.

Margins Margins in the specified unit of measure (shared with the Pixel Size combos) in X and Y directions that the surface should be extended beyond the bounding box of the objects or records being pasted to create the surface. This allows extending the created surface by some margins beyond the data being pasted to create the surface, allowing some interpolation at the edges for smoother edge effects in some cases. 0 by default.

Pixel Size Size of pixels in the created surface, in the specified unit of measure.

Same size in X and Y direction If checked (default), any edits in the X pixel size will also update the Y pixel size, for "square" pixels.

Method Gravity, Kriging, Median-polish Kriging, No Interpolation, Triangulation (flat) or Triangulation adjusted for contours. The type of interpolation algorithm (model) used to create the surface. Options other than Kriging will appear only if the optional Surface Tools extension has been installed. The Triangulation adjusted for contours method uses triangulation adjusted using the DEST algorithm for increased accuracy.

Neighbors Appears when using the Gravity, Kriging or Median-
Surfaces and Terrains

**polish Kriging** method. The number of neighboring points to consider when making the interpolation. Manifold will automatically use all points as neighbors for interpolations up to 1000 points. Note that it is faster to use all points as neighbors up to 1000 points than it is to use some subset of points (such as, only 990 neighbors out of 1000 points).

**Model**
Appears when using the Kriging or Median-polish Kriging method.

- **Auto** (default), Exponential, Gaussian, Linear, Power, Rational or Spherical interpolation models. The Auto setting allows Manifold to choose which of the interpolation models it thinks will work best in this case.

- **Linear, Power, Rational or Auto** choices will appear only if the optional Surface Tools extension has been installed.

**Use only Voronoi neighbors**
Interpolate over each location using only those neighbors that would be adjacent in a Voronoi diagram. This helps achieve a balance between a too-smooth interpolation achieved with a high number of neighbors and a too-coarse interpolation achieved with a low number of neighbors. Using this option will increase processing time.

**Use radius**
Appears when using the Gravity, Kriging or Median-polish Kriging method. The distance in the given units from any data point over which an interpolation can extend. Not checked by default to allow Manifold to choose the radius.

**Save error surface as**
Appears when using the Kriging or Median-polish Kriging method.

Allows saving of an error surface using the given numeric type.

**Set corner values to**
Appears when using the Triangulation (flat) or Triangulation adjusted for contours methods.

Optionally set the values in the corners of the computed surface to the given values. If this option is used, the entire surface (to the full rectangular extents) will be interpolated using the corner values as anchors. If this option is not used, those regions outside the convex hull of the existing data points will remain invisible.

**Pasting Drawings as Surfaces**

Pasting a drawing as a surface is similar to pasting a table as a surface except that there is no need to specify the fields to be used as latitude and longitude, since the coordinates of objects in the drawing are built into the geometry of the drawing.

Likewise, there is no need to specify whether the coordinates are latitude and longitude coordinates, since the coordinate system used by the drawing is also known. When pasting a drawing as a surface, the surface will inherit all projection parameters from the drawing.

When pasting a drawing as a surface, Manifold can use point and line objects to create a surface. Point objects are taken as point data from which the surface is interpolated. Line objects will have point data created at each coordinate ("inflection point") that defines the line.

**Height**
Choose the field to be used for elevation values for the surface.
Type
The data type to use for elevation values in the surface. If the originating field is a different type, Manifold will convert to the given type when writing to the surface.

Margins
Margins in the specified unit of measure (shared with the Pixel Size combos) in X and Y directions that the surface should be extended beyond the bounding box of the objects or records being pasted to create the surface. This allows extending the created surface by some margins beyond the data being pasted to create the surface, allowing some interpolation at the edges for smoother edge effects in some cases. 0 by default.

Pixel Size
Size of pixels in the created surface, in the specified unit of measure.

Same size in X and Y direction
If checked (default), any edits in the X pixel size will also update the Y pixel size, for “square” pixels.

Method
Gravity, Kriging, Median-polish Kriging, No Interpolation or Triangulation (flat). The type of interpolation algorithm (model) used to create the surface. Options other than Kriging will appear only if the optional Surface Tools extension has been installed.

Neighbors
Appears when using the Gravity, Kriging or Median-polish Kriging method. The number of neighboring points to consider when making the interpolation. Manifold will automatically use all points as neighbors for interpolations up to 1000 points. Note that it is faster to use all points as neighbors up to 1000 points than it is to use some subset of points (such as, only 990 neighbors out of 1000 points).

Model
Appears when using the Kriging or Median-polish Kriging method. Choose Auto (default), Exponential, Gaussian, Linear, Power, Rational or Spherical interpolation models. The Auto setting allows Manifold to choose which of the interpolation models it thinks will work best in this case. Linear, Power, Rational or Auto choices will appear only if the optional Surface Tools extension has been installed.

Use only Voronoi neighbors
Interpolate over each location using only those neighbors that would be adjacent in a Voronoi diagram. This helps achieve a balance between a too-smooth interpolation achieved with a high number of neighbors and a too-coarse interpolation achieved with a low number of neighbors. Using this option will increase processing time.

Use radius
Appears when using the Gravity, Kriging or Median-polish Kriging method. The distance in the given units from any data point over which an interpolation can extend. Not checked by default to allow Manifold to choose the radius.

Save error surface as
Appears when using the Kriging or Median-polish Kriging method. Allows saving of an error surface using the given numeric type.

Set corner values to
Appears when using the triangulation method. Optionally set the values in the corners of the computed surface to the given values. If this option is used, the entire surface (to the full rectangular extents) will be interpolated using the corner values as anchors. If this option is not used, those regions outside the convex hull of the existing data points will remain...
Trimming Unwanted Surface Extensions

Data pasted as a surface rarely fits into a rectangular box. Normally, the interpolation radius parameter will deal with this automatically. For example, suppose we have a drawing showing county centroids in the United States with a "population" field for each county. If we paste these as a surface, the data within the borders of the US will be real but there will be no data beyond the borders of the US. In that case, the interpolation radius will render invisible all pixels in the surface beyond the region where data points exist. When using the triangulation method, the **Set corner values** option allows us to force Manifold to fill in all invisible pixels outside the convex hull of the surface to form an interpolated rectangular surface.

Another option is to simply select the unwanted surface pixels and delete them. The easiest way to make a surface conform to some desired area in a drawing is to select the area and transfer the selection to the surface. Use **Edit - Invert Selection** to select all other pixels and delete them. Next, use the Selections pane to select invisible pixels, invert the selection once more and then use the transform toolbar to **Crop** to the selection.

Creating Tables from Surfaces

Surfaces may be copied and then pasted as tables. See the Editing Surfaces topic.

Unexpected Results in Surfaces

When creating surfaces from tables or drawings at times we may see unexpected results in the surface. For example, a regular pattern or other visual artifact might appear superimposed upon the surface as seen in the illustration below.

The illustration above shows a surface interpolated from a table of 641 records providing elevations near Portola Valley, California. The table was copied and then pasted using default settings for pixel size into a surface. Using default resolution creates a surface with 102,400 points which means that 99.4% of the points are interpolated using a Kriging algorithm.

For smoother results it would be better to paste the table using larger pixels, for, say, 1600 points with approximately 60% being interpolated…

…or perhaps somewhat smaller pixels to get 6400 points with approximately 90% being interpolated:
Both of the above illustrations show the full surface, of which the first illustration shows the upper right corner. Pasting tables or drawings as surfaces where the surface contains much more than 90% interpolated points will produce highly unreliable results.

It is always tempting for newbies to want to interpolate data to "fill in the gaps" between a data set that has too few data points. It is also easy to forget the simple arithmetic that creates from such innocuous sounding numbers as a 320 x 320 (X pixels by Y pixels) the huge number of 102,400 points. Between the desire to have data where there is none and a forgetfulness about how many pixels are involved in even medium-resolution surfaces it is easy to fall into the trap of trying to stretch a rather small data set into a very large surface. If we want to create a finely detailed, high-resolution surface from a table the only solution is to have a reasonably large number of data points (records) with which to begin.

**Example**

In this example we will create a surface from a grid of points.

We begin with a map that shows a drawing containing a grid of points superimposed upon the example Montara Mountain surface (created from the USGS 1:24K-scale SDTS DEM showing a region just south of San Francisco, California). The grid was created by creating a blank drawing and then using the Grid command to create a grid of points spaced every 1000 meters. Points in the grid that did not appear above the surface were deleted. Using the Transfer Heights command we placed the value of the height of the surface into the Height column for each point.

We now have a drawing that contains points with height values for each point. Many data sets from which we would like to create a surface will be in this form, as a drawing of points where each point has a height value. Although many such data sets will have the points in a regular arrangement, a regular arrangement of points is not necessary to create a surface from such points. In this example, the points are regularly arranged because
that was the easiest way to create points for the example, by using the Grid command. All that is necessary to create a surface from points is that the points are reasonably dense in areas of interest. Obviously, if some regions have very many data points while in other regions there are very few points then the accuracy with which a surface may be interpolated will be lower in regions with fewer points.

To create a surface, in the project pane we Copy the drawing containing points and Paste As a surface.

In the Paste As Surface dialog we choose to make each pixel 100 meters in size in X and Y directions. Since our original data set had a point every 1000 meters, this means that we will create a surface for which ten surface pixels will be interpolated between each data point. We have also checked the Save error surface box to create an error surface.
Viewing the result, we can see that the surface is clearly of much lower resolution than the original. Even so the surface reproduces the general topography, at least as much as can be represented given that the surface heights were sampled only every 1000 meters (the spacing of the grid points). Note that the only way to completely reproduce the original surface would have been to lay down a grid of so many points that each pixel of the original surface could be sampled.
Opening the error surface we can see the regions of higher error by applying the spectrum palette in the View - Display Options dialog, arranged so that regions of higher error show red and those of lower error show blue and purple. Note that the regions immediately surrounding the data points used to interpolate the surface show relatively low error while those regions surrounding rapidly changing terrain elevations (such as the higher altitude peaks) show greater error.

**Notes**

Images that represent raster data such as elevations, temperatures and other factors are often very interesting to visualize as surfaces.

Because Manifold can paste tables directly into surfaces, one can take a table of elevations created from, say, a GPS and paste it immediately as a surface. One could also take a table of any value (such as pollution levels, temperature, etc.) and paste it as a surface.

If a field named `z`, `elevation`, `elev`, `e`, `height`, `hgt` or `h` (case insensitive) exists in the drawing or table, Manifold will offer that by default as the field in the **Height** box. When pasting tables, fields named `longitude`, `latitude`, `lon`, `long`, `lat`, `x` or `y` (case insensitive) will be offered as the **X / long** or **Y / lat** box fields by default.

Attempting to create a surface larger than 16 million pixels (4,000 x 4,000) will trigger a confirmation dialog that will ask "Surface size is (x dimension) x (y dimension) pixels. Paste anyway?"

**See Also**

See the Displaying Data in a Gradient Map topic for an example that shows demographic data displayed in a continuously varying gradient throughout the United States using a surface.
Transform Toolbar - Surfaces

The Transform toolbar makes changes throughout the entire surface. If a selection is present it will operate only on those pixels in the selection. If no selection is present it will apply to all pixels in the surface.

The Transform toolbar consists of three boxes, from left to right: a target box, the operator box, and a parameter box.

- **Target box**: The pixels that will be affected, altered or which will control the operation. The target box will be pre-loaded with either [All Pixels] or with [Selection] depending on whether a selection is present or not.

- **Operator box**: The function to be applied. The operator box is context sensitive and will show only those operations that make sense for surfaces.

- **Parameter box**: The value to be used. Depending on the operator, this may be another pixel set or a value entered by the user. Many operators, such as Crop, do not require a parameter. The parameter box will not be enabled for such operators.

We use dark blue, black, and violet bold face fonts in this topic to distinguish the three boxes to make examples and explanation more clear. In real life Manifold uses the same black font color in all three Transform toolbar boxes.

**Using the Transform toolbar**

1. Click on the map layer or surface that contains the target pixels.
2. Make a selection if the operation is to be applied just to the selection.
3. Choose the desired operator in the operator box.
4. Choose or specify a value in the parameter box, if this operation requires it.
5. Press Apply.

**Target box**

The left-most box specifies the target pixels. These are the pixels that will be affected, altered or which will control the operation. The example above shows that there is no selection present, so the Transform toolbar will apply the Transform function to all of the pixels in the surface. If we make a selection, the target box would switch to [Selection] to show that whatever operation we apply with the Transform toolbar will act only on those pixels that are part of the current selection.

The target box is not enabled for changes when used with surfaces because the scope of the operation (All Pixels or Selection) is automatically determined by whether there is a selection present or not. When used with Tables, where the scope (which fields) can be user selected, the target box will be enabled for changes.

**Operator box**
Choose a function from the long list of operators available in the operator box. The example shows Add Noise, which will add random noise pixels to the surface. Note that many operators in the Transform toolbar's operator box are simplified versions of more flexible operators in the Surface menu. The Transform Operators - Surfaces topic lists operators available for use with surfaces. Many of these operators are the same as or directly analogous to transform operators used with images.

**Parameter box**

Many operators do not require any parameters. For these functions the parameter box will be disabled. Other functions will require specification of an additional parameter. This may be another pixel set (such as [All Pixels], [Selection] or the names of saved selections), or more frequently it could be a parameter that is entered by the user. To enter a parameter, click into the parameter box, enter the value using the keyboard and then press Enter.

**Examples**

[All Pixels] Rotate -90 - Rotate the surface counter-clockwise 90 degrees.

[Selection] Crop - Crop the surface to the minimum enclosing rectangle of the selection.
Transform Operators - Surfaces

Many transform operators used with surfaces are the same as or directly analogous to transform operators used with images. See the Transform Operators - Images topic for a guide to operators used in images, including links to topics that illustrate many of the concepts used in transforms that are similar to those in surfaces. The hyperlinks in the table below will usually jump to the images version of the command.

- **Add Margin** Add to the rectangular size of the surface by the given number of pixels at the margin.
- **Blur** Blur image by fixed amount. Blurs both features and elevations. For a "blur" effect that does not mix elevations, use the Median filters.
- **Blur (parameter)** Blur image by the number of pixels given in parameter. This blur is implemented as a matrix operator where the value of the parameter is the weight of the center pixel with weights of all other pixels set to 1. Increasing the parameter therefore produces a lesser blur.
- **Crop** Normally used with [Selection] as the scope. Crop surface size to the selection.
- **Crop Margin** Crop the rectangular size of the surface by the given number of pixels at the margin.
- **Difference East** Detect and emphasize edge transitions to the East. See the Filters topic for general discussion of convolution matrix filters like this one.
- **Difference North** Detect and emphasize edge transitions to the North.
- **Difference North-East** Detect and emphasize edge transitions to the North-East.
- **Difference North-West** Detect and emphasize edge transitions to the North-West.
- **Difference South** Detect and emphasize edge transitions to the South.
- **Difference South-East** Detect and emphasize edge transitions to the South-East.
- **Difference South-West** Detect and emphasize edge transitions to the South-West.
- **Difference West** Detect and emphasize edge transitions to the West.
- **Flip Horizontally** Flip surface left / right to mirror surface.
- **Flip Vertically** Flip surface top / bottom to upside down surface.
- **High Pass 1** Emphasize rapid transitions between pixels. Good edge detection with horizontal and vertical lines.
- **High Pass 2** Stronger emphasis on rapid transitions between pixels with strong edge detection on lines of all angles.
- **High Pass 3** Edge detection with crisp vertical and horizontal and fuzzy lines at other angles.
- **Interpolate** Fill in invisible pixels by interpolation. Set the invisible pixel to the average value within the smallest rectangular box centered at given pixel that contains at least one visible pixel. If all pixels in the surface are invisible, the operator makes no changes.
- **Interpolate (Parameter)** Fill in invisible pixels by interpolation. Set the invisible pixel to the average value within the rectangular box of a given radius centered at given pixel, and leave the pixel invisible if the box contains no visible pixels. Depending on the size of the parameter, this operator can be extremely slow although if the radius parameter is chosen to incorporate many pixels it is the most accurate. Slight increases in the radius parameter can...
result in geometrically slower operation, so scale up carefully when using this operator.

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interpolate Row</td>
<td>Fill in invisible pixels by interpolation. Set the invisible pixel to the weighted average of the visible pixels to the left and right of the given pixel in the same row, and leave the pixel invisible if its row does not contain any visible pixels. This operator is faster but not as accurate as the other interpolation operators.</td>
</tr>
<tr>
<td>Invert</td>
<td>Invert pixel values so that 0 becomes 255 and 255 becomes 0.</td>
</tr>
<tr>
<td>Laplace 1</td>
<td>Good edge detection with emphasis on horizontal and vertical changes.</td>
</tr>
<tr>
<td>Laplace 2</td>
<td>Strongest changes at point of intersection of horizontal and vertical lines, resulting at a small X pixel pattern at the point of intersection; fuzzy detection of linear features at other angles.</td>
</tr>
<tr>
<td>Low Pass 1</td>
<td>Strong blur, with strong suppression of large pixel to pixel changes.</td>
</tr>
<tr>
<td>Low Pass 2</td>
<td>Blur, with medium suppression of large pixel to pixel changes.</td>
</tr>
<tr>
<td>Low Pass 3</td>
<td>Blur, with least suppression of large pixel to pixel changes. Removes graininess in some surfaces.</td>
</tr>
<tr>
<td>Median Cross</td>
<td>Evens out elevation tending to preserve horizontal and vertical features. A &quot;blur&quot; that ignores perfectly horizontal and vertical linear features.</td>
</tr>
<tr>
<td>Median Square</td>
<td>Evens out elevation looking at a 3 x 3 matrix surrounding each pixel. A &quot;blur&quot; effect that preserves prevailing elevation.</td>
</tr>
<tr>
<td>Median Square 5</td>
<td>Strong evening out of elevation looking at a 5 x 5 matrix surrounding pixels of elevation differing greatly from surrounding pixels. A stronger &quot;blur&quot; effect that preserves prevailing elevation.</td>
</tr>
<tr>
<td>Rotate</td>
<td>Rotate surface given number of degrees. Negative degrees are taken as counter-clockwise rotation.</td>
</tr>
<tr>
<td>Sharpen</td>
<td>Emphasize transitions to give images a sharper appearance.</td>
</tr>
<tr>
<td>Sharpen (parameter)</td>
<td>A Sharpen with selectable parameter value for degree of sharpness. This sharpen is implemented as a matrix operator where the value of the parameter is the weight of the center pixel with weights of all other pixels set to 1. Increasing the parameter therefore produces less sharpening.</td>
</tr>
<tr>
<td>Sharpen More</td>
<td>Enhanced sharpness compared to standard Sharpen.</td>
</tr>
<tr>
<td>Threshold Lower</td>
<td>Resets all pixel values below the elevation given in the parameter to that elevation value.</td>
</tr>
<tr>
<td>Threshold Upper</td>
<td>Resets all pixel values above the elevation given in the parameter to that elevation value.</td>
</tr>
<tr>
<td>Tile</td>
<td>Divides surface into square tiles of given extent and then averages pixel values within the tiles. This is used for aggregating and interpolating surfaces into larger tiles and for creating &quot;pixelated&quot; artistic effects.</td>
</tr>
</tbody>
</table>
| Tile Median       | Equivalent to Tile command, but uses a median computation to derive the elevation to be used from the tile from only those elevations that already exist in the surface. Useful to assure elevations of tiles will be round numbers if only round numbers had previously existed in
the surface.
Transform - Surface - Interpolate Operators

The interpolation transform operators discussed in this topic are enabled when the optional Surface Tools extension has been installed and the focus is on a surface window. If you have not activated the Surface Tools extension with a valid serial number the capabilities in this topic will not be available.

Manifold provides three interpolation operators that are used to fill in invisible pixels within surfaces.

- **Interpolate** sets the invisible pixel to the average value within the smallest rectangular box centered at given pixel that contains at least one visible pixel. If all pixels in the surface are invisible, the operator makes no changes.

- **Interpolate (Parameter)** sets the invisible pixel to the average value within the rectangular box of a given radius centered at given pixel, and leaves the pixel invisible if the box contains no visible pixels. Depending on the size of the parameter, this operator can be extremely slow. Slight increases in the radius parameter can result in geometrically slower operation, so scale up carefully when using this operator.

- **Interpolate Row** sets the invisible pixel to the weighted average of the visible pixels to the left and right of the given pixel in the same row, and leaves the pixel invisible if its row does not contain any visible pixels. This operator is faster but not as accurate as the other interpolation operators.

Note: The Interpolate transform usually provides the best results overall. The Interpolate (Parameter) transform has been provided for the use of experts who want precise control of the interpolation process.

**Example**

Suppose we begin with a surface that has some missing pixels, represented by invisible pixels.

Normally, of course, it would be unusual to have a surface with such relatively large swaths of missing pixels. The usual case is to have a few pixels missing, perhaps as a result of noise.
The **Interpolate** transform operator does a good job of filling missing pixels. The smaller the region of invisible pixels the better the result.

Using **Interpolate (Parameter)** with a parameter of 50 (meaning a radius of 50 meters) results in an overly-even interpolation.

The **Interpolate Row** operator results in lengthwise striations since it interpolates by row. This command is very fast, but it is best used in "flat" regions.
**Tech Tip**

The **Interpolate** transforms operate on selected pixels, but both selected and unselected pixels will be used to compute the interpolation required for the effect upon selected pixels if unselected pixels also are within range of the interpolation.
Transform - Surface - Tile Operators
Divides surface into square tiles of given extent and then averages pixel values within the tiles. This is used for aggregating and interpolating surfaces into larger tiles and for creating "pixelated" artistic effects.

Example

The surface on the left was tiled into the surface on the right using tiles of 10-pixel size.

Setting up the transform toolbar as seen above and pressing Apply will tile the surface into tiles of 10-pixel extent. Note that if the accompanying terrain is open it too will be tiled.

Before tiling the terrain is seen from the position marked in the surfaces above.
After tiling the terrain is rendered based on the square, even tiles that now make up the surface.

Comments

The terrain used is the Montara Mountain, California, 1:24K-scale SDTS DEM downloaded from USGS. The view in the terrain shows San Andreas lake in the valley of the San Andreas Fault just south of San Francisco.

Tile will find exact mathematical values for the elevation of each tile. These will often not be round numbers. If the surface originally used elevations that are round numbers (such as whole feet or meters) use Tile Median to preserve such round number values.

Surface Tools

Surface Tools

The Surface Tools package is an optional extension to Manifold System that provides additional commands for working with surfaces. The extension adds functionality in several areas:

- The Surface - Transform command dialog allows arbitrary transformation of surfaces, including computations that involve multiple surfaces such as subtracting one surface from another. A very rich collection of operators allows many different types of tasks to be accomplished.
- The Surface - Watersheds command works with a surface and optionally a drawing to find watersheds in the surface (regions sharing a common drainage) as well as streams in the surface or upstream areas in the surface from points in a specified drawing.
- The Transfer Heights command enables rapid transfer of surface values from surfaces to points.
- The Visible Area command shows areas that are visible from given points. The height of viewing points may be automatically adjusted from a height field.
- New options in surface generation allow creating surfaces from drawings and tables using Gravity interpolation, Median-polish Kriging and triangulation. See the Project Pane - Paste as Surface topic.
- Expanded model choices for Krige interpolation include Linear, Power, Rational and automatic model choices.
- New transform toolbar operators for interpolation of missing pixels, Interpolate, Interpolate (Parameter) and Interpolate Row. See the Transform - Surface - Interpolate Operators topic for details.
- Surface Tools adds the ability to work with new Profiles and Elevations components that show cross-sectional cuts through a given path over a surface.
- Expanded programmatic access to surface operations.
If you do not have the optional **Surface Tools** package enabled you will not have the above capabilities enabled within Manifold System. This documentation describes these commands but they will not be enabled if you do not have the **Surface Tools** package enabled.

### Installing Surface Tools

The **Surface Tools** package is built into your Manifold System installation and becomes enabled for use when activated. Activate it by providing a **Surface Tools** serial number and Activation key that turn on **Surface Tools** functionality. The process of activating the **Surface Tools** extension is very similar to that used to activate Manifold System. See the Activation Keys and Serial Numbers topic and the Installing and Activating a Manifold Extension topic for details.

Some Manifold product options, such as Universal Edition or Ultimate Edition, use a single serial number and Activation key to enable permanent installation of both Manifold System as well as extensions. If you have activated Universal Edition or Ultimate Edition you do not need to individually turn on **Business Tools** or **Geocoding Tools** or **Surface Tools**. Activating Universal Edition or Ultimate Edition will also automatically turn on all three extensions.

### Turning on Surface Tools

If you have licensed Universal Edition, you do not need to turn on **Surface Tools**. The **Surface Tools** extension is automatically enabled with a Universal Edition serial number.

If you have not licensed Universal Edition and you wish to add **Surface Tools** functionality to your installation, begin by acquiring a **Surface Tools** license from manifold.net, which will provide you with a **Surface Tools** serial number.

2. Launch Manifold System. Close any projects that may be open.
3. Choose Help - Activate Extension
4. Enter the **Surface Tools** serial number. Enter it exactly as it was issued to you by manifold.net. Do not change upper case to lower case. Do not replace hyphens with space characters or make any other changes to the serial number. If desired, you may also enter an Activation key for the serial number as well. Press Accept.
5. Manifold will then exit. When re-launched, the Surface Tools package will be enabled.
6. Using the serial number alone the **Surface Tools** package may be run for 30 days from the date the serial number was issued. After that, an Activation Key must be fetched and provided to the Help - Activate Extension dialog together with the serial number. Once an Activation Key and a serial number have been provided to the Help - Activate Extension dialog the extension will be permanently installed. Don't forget to login as Administrator when permanently activating the **Surface Tools** extension.

### Status of Extensions

The Help - About dialog shows all extensions that have been installed. If a serial number only has been used to install an extension, the number of days left before an Activation key is also required will be shown.

### See Also

Transforming Surfaces
- Transform Dialog Functions and Operators
- Transform - Surface - Interpolate Operators
- Surface - Transform
Transfer Heights
Project Pane - Paste as Surface
Profiles and Elevations
Drawing - Visible Area
Activate Extension
Activation Keys and Serial Numbers
Transforming Surfaces

Surfaces may be transformed in Manifold using several main methods:

- Transform Toolbar - The transform toolbar may be used for simple, one-line manipulation of surfaces. The transform toolbar for surfaces is available in all Manifold editions. The optional Surface Tools extension adds new interpolation commands discussed in the Transform - Surface - Interpolate Operators topic.
- Surface - Transform Dialog - The Transform dialog allows sophisticated manipulation of surfaces via the evaluation of formulas. Formulas can be simple expressions, such as adding one surface to another, or they can include sophisticated program logic using a host of different operators. The Transform dialog is available only if the optional Surface Tools extension has been installed.
- Surface menu commands - Commands such as Surface - Resize may be used to alter surfaces.
- Scripting - By writing scripts or other programs surfaces may be changed.

This topic discusses operations using the Surface - Transform dialog. The Surface - Transform menu selection is enabled when the optional Surface Tools extension has been installed and the focus is on a surface window. If you have not activated the Surface Tools extension with a valid serial number you will not be able to use the Surface - Transform command and the capabilities in this topic will not be available.

Many functions in the Surface - Transform dialog can use NVIDIA CUDA for nearly instantaneous computation if our systems are so equipped. See the NVIDIA CUDA topic for details.

The Surface - Transform Dialog

![Surface - Transform Dialog](image)

To launch the Transform dialog, open the surface that is to be transformed and choose Surface - Transform in the main menu. The Transform dialog allows us to choose the scope in the upper combo box and to enter a formula into the lower pane. The dialog also includes a Surfaces pane that lists all surfaces in the project. We can double click on a surface in the list to add it to the Formula pane and thus save ourselves the effort of manually entering the name of a surface.

The scope specifies the pixels that will be affected by the transformation, and can be [All Pixels], [Invisible Pixels] or any selection or saved selection made in the surface. Formulas can contain numbers, names of surfaces (names of surfaces must be enclosed in square brackets [] if the name contains a space character), arithmetic operations and functions. Formulas are case insensitive. When we press the OK button, each pixel in the surface in the scope will be replaced with the value of the formula for that pixel.

If we would like to preserve the original surface unmodified we can check the Save result as new component box and the system will create a new surface (a copy of the surface which had the focus when the dialog was launched) and put the results of the formula there. The Save result as new component also allows us to
surface transformations on a surface that is a read-only component, such as perhaps a surface that has been linked into a project from an Enterprise storage using the Enterprise Edition of Manifold System.

Suppose we have three surfaces named A, B and C. If we open A and invoke Surface - Transform we can enter

\[ A + B \]

into the Formula pane and press OK. This will set all heights in A to the sum of A + B. If we had entered a formula using the Max() function (one of many Transform Dialog Functions and Operators that may be used) as follows:

\[ \text{Max}(A, B + C + 1) \]

we would have set the height of each pixel in A to either the value of that pixel in A or the value of B + C + 1, whichever is larger.

When evaluating formulas, Manifold automatically handles coordinate system conversions so that a surface in lat/lon and a surface in UTM projection may be used together in a formula. Manifold will also automatically handle type conversions so that a surface with floating-point values for height may be used in a formula together with a surface using integer values for heights.

Invisible pixels are also automatically handled. A formula consisting of just "A" makes all pixels that are visible in A visible in the result, and it makes all pixels that are invisible in A invisible in the result. The formula "A + B" makes all pixels that are visible in both A and B visible in the result, and makes all pixels that are invisible in either A or B invisible in the result.

**Example**

Suppose we have two surfaces that show the intensity of the downwind chemical plumes from two different releases of a hazardous chemical:

Suppose the releases happened on different days from different positions so that because of changes in wind and release position and amount of chemical released the plumes have different orientations and show different intensities. The two surfaces are called Plume A and Plume B. These surfaces are correctly georegistered to a true location on Earth, so that the inclined rectangle that is the area of interest is exactly at the same location for both surfaces.
Dark green shows zero chemical detected, while brighter green shading to beige to darker colors to white shows increasing levels of exposure. Our task is to sum the two surfaces so we can see what are the values of combined exposure to the two plumes in the area of interest.

This is easy. We first duplicate Plume A and rename it to Combined Plumes. We do this so that the original surface will not be altered when we do a transform.
With the focus on the Combined Plumes surface, we choose Surface - Transform and enter

\[[\text{Plume A}] + [\text{Plume B}]\]

into the Formulas pane. We can save time keyboarding by double clicking on the names of the surfaces in the Surfaces box to add them to the formula instead of typing out them out manually. We press OK.

The surface is transformed so that each pixel now has a value that is the value of that location in Plume A plus the value of that location in Plume B. We can immediately see where the cumulative effects of the plumes are creating “hot spots” not immediately obvious from a visual examination of only Plume A or Plume B.
Note that the pixels that were invisible in both Plume A and Plume B continue to be invisible in the combined result. If we turn on the Layers pane and turn on the Border we can see that a considerable part of the surface is invisible pixels.

Can we use the Transform dialog to make these invisible pixels visible? That also is easy.

We launch the Transform dialog once more and choose [Invisible Pixels] as the scope. In the Formula pane we enter 0. This terse expression simply means to put the value zero into all pixels in the scope, that is, into all pixels that are invisible pixels.
When we press OK the invisible pixels will be filled in with the value 0 so they will be colored the same dark green as all the other zero pixels.

If we combine the surfaces in a map with a drawing of roads we can see where the chemical plumes have combined to generate regions of higher exposure. The illustration uses partial transparency in the surface layer.
We can also add invisible pixels using the **Null** constant with an expression like that above, which converts all pixels with a value of 100 or less into invisible pixels.

The bright green zone in the combined plumes corresponds to a value of **100 so the formula used renders invisible all pixels except those in the plumes themselves, stripping out the zero-valued dark green pixels.**
This may be a more useful effect when combined with other surfaces in a map. The map above shows the surfaces using partial Layer Opacity for each for a better effect.

This is a somewhat contrived example, but it shows the ease with which the **Surface - Transform** dialog may be used to perform calculations using multiple surfaces.

**Example**

The **Surface - Transform** dialog may be used for a variety of utility operations, such as interpolating regions of missing (invisible) pixels.

Suppose we have a surface called **Montara** that shows the terrain in the region of Montara Mountain, California. The surface has portions with no data, represented with invisible pixels.
We may use the Surface - Transform dialog with the Interpolate(s) function to fill in the missing pixels as seen above. Note that the name could have been cited as simply (Montara) because there are no spaces in the name. However, the example uses ([Montara]) with square brackets [ ] because the name was added by double clicking on the surface name in the Surfaces pane to save having to manually enter the name.

The result is imperfect, but better than having invisible pixels. See the Interpolate Operators topic for additional examples.

Example

Open the Montara Mountain sample surface. Launch the Surface - Transform dialog, check the Save result as new component option and run the following formula:

```
FlowAccum(FlowDir(FillSinks([Montara])))
```

This will create a flow accumulation surface. Rename the resulting surface Montara Flow.
The following query counts the number of pixels with flow exceeding 10000:

```sql
SELECT Count(*) FROM [Montara Flow] WHERE [Height (I)] > 10000;
```

The following query finds the coordinates of the top 10 maximum flow pixels:

```sql
SELECT TOP 10 * FROM [Montara Flow] ORDER BY [Height (I)] DESC;
```

See the Watersheds topic for additional information on flows.

**Available Expressions in Formulas**

Formulas may use +, -, *, / as arithmetic operators (addition, subtraction, multiplication and division). Pi (3.1415926...) and Null are the only available constants. Null is used to create invisible pixels.

A vast number of functions and operators are available, and are listed in the Transform Dialog Functions and Operators topic.

**Merging Surfaces**

It may be tempting to use the Surface - Transform tool to attempt to merge two or more overlapping or non-overlapping surfaces into a single surface. However, this is not something done with the Surface - Transform tool. The Surface - Transform tool takes a surface and sets its pixels to values possibly taken from other surfaces, and then perhaps saves the result as a new surface. It never attempts to enlarge or reduce the surface.

For example, given two surfaces A and B where B is a larger surface that partially overlaps A, the expression A + B operating on surface A will not expand the bounding box of A to include a larger B surface. It will simply set the pixels of the resulting surface (which will be the same size as A) to the sum of values in A and B and make pixels at locations that do not belong to either A or B invisible.

To merge a pair of surfaces together, select all pixels in the first surface, copy them and paste the pixels into the second surface. Make sure the Confirm expanding images or surfaces when pasting option in the Tools - Options - Miscellaneous dialog is turned on.

**Short Examples**

Given surfaces A, B and C with surface A in feet and surfaces B and C in meters, we may execute the following formulas on surface A:

```sql
A + 1
```

Increases the height of all visible pixels in A by 1.

```sql
A * 0.3048
```

Converts the heights of all visible pixels in A from feet to meters.

```sql
IIf(B > 2000, Max(A, B/0.3048), A)
```

Makes pixels in A at least as high as those in B in regions where the height of B exceeds 2000 meters.

```sql
LowPass1(B) + HighPass1(C)
```

Creates a composite surface (in meters) taking low frequencies from B and high frequencies from C.

```sql
A + B/0.3048
```
Increases the height of all visible pixels in A by the height of the respective pixels in B. Makes pixels in A that are invisible in B invisible.

\[
\text{IIf(HasValue(B, 0, 0), A + B/0.3048, A)}
\]

Increases the height of all visible pixels in A by the height of the respective pixels in B. Leaves pixels in A that are invisible in B intact.

\[
\text{IIf(A < 1000, Null, A)}
\]

Makes all pixels below 1000 invisible and leaves all other pixels intact.

**Tech Tips**

- In case of disaster, single-step **Undo** works to undo the results of the **Transform** dialog.

- The **Transform** dialog remembers the previous formula. This is very useful together with **Undo** to make adjustments to formulas that create unexpected results.

**See Also**

- NVIDIA CUDA
- Surface Tools
- Transform Dialog Functions and Operators
- Transform - Surface - Interpolate Operators
Transform Dialog Functions and Operators

The functions and operators discussed in this topic are available in the Surface - Transform dialog's Formula pane. The Surface - Transform menu selection is enabled when the optional Surface Tools extension has been installed and the focus is on a surface window. If you have not activated the Surface Tools extension with a valid serial number you will not be able to use the Surface - Transform command and the functions in this topic will not be available for use with that command.

Surface - Transform functions taking a drawing name as an argument will also accept the names of drawing tables or theme components.

Many functions in the Surface - Transform dialog can use NVIDIA CUDA for nearly instantaneous computation if our systems are so equipped. See the NVIDIA CUDA topic for details.

Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abs(x)</td>
<td>Returns the absolute value of x.</td>
</tr>
<tr>
<td>ACos(x)</td>
<td>Returns the arccosine of x in the range of 0 to pi radians.</td>
</tr>
<tr>
<td>ASin(x)</td>
<td>Returns the arcsine of x in the range of -pi/2 to pi/2 radians.</td>
</tr>
<tr>
<td>Aspect(s, w)</td>
<td>Returns the aspect of s in the range of -180 to 180 degrees, computed over a window of specified size (1 for 3x3, 2 for 5x5, etc). CUDA enabled.</td>
</tr>
<tr>
<td>Atn(x)</td>
<td>Returns the arctangent of x in the range of -pi/2 to pi/2 radians.</td>
</tr>
<tr>
<td>Atn2(y, x)</td>
<td>Returns the arctangent of y/x in the range of -pi to pi radians.</td>
</tr>
<tr>
<td>AvgValue(s, w)</td>
<td>Returns the average value of s in a window of specified size (1 for 3x3, 2 for 5x5, etc). CUDA enabled. With CUDA, the maximum window size is 31.</td>
</tr>
<tr>
<td>Blur(s)</td>
<td>Returns the value of the Blur filter. CUDA enabled. Example: Blur([Plume A])</td>
</tr>
<tr>
<td>Ceil(x)</td>
<td>Returns the smallest integer that is greater than or equal to x.</td>
</tr>
<tr>
<td>Cos(x)</td>
<td>Returns the cosine of x.</td>
</tr>
<tr>
<td>CurvGauss(s, w)</td>
<td>Returns the Gaussian curvature of s, computed over a window of specified size (1 for 3x3, 2 for 5x5, etc). CUDA enabled.</td>
</tr>
<tr>
<td>CurvMean(s, w)</td>
<td>Returns the mean curvature of s, computed over a window of specified size (1 for 3x3, 2 for 5x5, etc). CUDA enabled.</td>
</tr>
<tr>
<td>CurvPlan(s, w)</td>
<td>Returns the plan curvature of s, computed over a window of specified size (1 for 3x3, 2 for 5x5, etc). CUDA enabled.</td>
</tr>
<tr>
<td>CurvProfile(s, w)</td>
<td>Returns the profile curvature of s, computed over a window of specified size (1 for 3x3, 2 for 5x5, etc). CUDA enabled.</td>
</tr>
<tr>
<td>Deg2Rad(x)</td>
<td>Converts x from degrees to radians.</td>
</tr>
<tr>
<td>DifferenceE(s)</td>
<td>Returns the value of the DifferenceE filter. CUDA enabled.</td>
</tr>
<tr>
<td>DifferenceN(s)</td>
<td>Returns the value of the DifferenceN filter. CUDA enabled.</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DifferenceNE(s)</td>
<td>Returns the value of the DifferenceNE filter. CUDA enabled.</td>
</tr>
<tr>
<td>DifferenceNW(s)</td>
<td>Returns the value of the DifferenceNW filter. CUDA enabled.</td>
</tr>
<tr>
<td>DifferenceS(s)</td>
<td>Returns the value of the DifferenceS filter. CUDA enabled.</td>
</tr>
<tr>
<td>DifferenceSE(s)</td>
<td>Returns the value of the DifferenceSE filter. CUDA enabled.</td>
</tr>
<tr>
<td>DifferenceSW(s)</td>
<td>Returns the value of the DifferenceSW filter. CUDA enabled.</td>
</tr>
<tr>
<td>DifferenceW(s)</td>
<td>Returns the value of the DifferenceW filter. CUDA enabled.</td>
</tr>
<tr>
<td>Diversity(s, w)</td>
<td>Returns the number of different values of s in a window of specified size.</td>
</tr>
<tr>
<td>DiversityIndex(s, w)</td>
<td>Returns the diversity index of s in a window of specified size as a value</td>
</tr>
<tr>
<td>Exp(x)</td>
<td>Returns the exponent of x.</td>
</tr>
<tr>
<td>FillSinks(s)</td>
<td>Fills depressions in s.</td>
</tr>
<tr>
<td>Fix(x)</td>
<td>Removes the fractional part of x rounding towards zero</td>
</tr>
<tr>
<td>Floor(x)</td>
<td>Returns the largest integer that is less than or equal to x.</td>
</tr>
<tr>
<td>FlowAccum(dir, den)</td>
<td>Computes flow accumulation using flow directions (dir) and optional flow</td>
</tr>
<tr>
<td>FlowAreas(dir, str)</td>
<td>Computes watersheds using flow directions (dir) and streams (str) by</td>
</tr>
<tr>
<td>FlowDir(s)</td>
<td>Computes flow directions for s. Returns true degrees between -180 to 180</td>
</tr>
<tr>
<td>FlowStream(dir, x, y)</td>
<td>Computes a stream starting at a given location using flow directions (dir),</td>
</tr>
<tr>
<td>FlowStreams(dir, acc, thr)</td>
<td>Computes streams using flow directions (dir) and flow accumulation (acc). Streams with accumulation lower than the threshold (thr) are ignored. Returns a unique stream ID for each pixel belonging to a stream whose accumulation is higher than or equal to the threshold, and 0 for all other pixels. Note that there are two variants of the FlowStreams function, one of which takes a drawing as an argument and one that does not.</td>
</tr>
<tr>
<td>FlowStreams(dir, drawing)</td>
<td>Computes streams for all points in a given drawing using flow directions (dir). Returns a unique stream ID for each pixel belonging to a stream, and 0 for all other pixels. Note that there are two variants of the FlowStreams function, one of which takes a drawing as an argument and one that does not.</td>
</tr>
<tr>
<td>FlowUpstreamArea(dir, x, y)</td>
<td>Computes an upstream area for a given location using flow directions (dir). Returns 1 for pixels belonging to the upstream area, and 0 for all other pixels.</td>
</tr>
<tr>
<td>FlowUpstreamAreas(dir,</td>
<td>Computes upstream areas for all points in a given</td>
</tr>
</tbody>
</table>
drawing) drawing using flow directions (dir). Returns the ID of the point object for each pixel belonging to its upstream area, and 0 for all other pixels.

Gravity(drawing, column, neighbors, model, voronoi) Interpolates objects in the given drawing using the given column as the height. Line objects participate as sets of their coordinates. Area objects are ignored. Setting the number of neighbors to -1 will use the default number of neighbors. The model argument should be a blank string. The voronoi argument is a boolean (if true, only Voronoi neighbors are used). All arguments except the first two are optional.

HasValue(s, dx, dy) Returns True if surface s has a visible pixel at current position offset by dx and dy and False otherwise.

Height(s) Returns the number of rows in surface s.

HighPass1(s) Returns the value of the HighPass1 filter. CUDA enabled.

HighPass2(s) Returns the value of the HighPass2 filter. CUDA enabled.

HighPass3(s) Returns the value of the HighPass3 filter. CUDA enabled.

Ilf(a, b, c) Returns b if a is True and c otherwise.

Interpolate(s, p) Returns the value of s at the current pixel if it is visible and the interpolated value if it is invisible. The interpolated value is computed as the average value of s in the smallest box centered at the current pixel that contains at least one visible pixel. If all pixels in s are invisible, the function returns Null. The optional parameter p allows specification of an interpolation radius over which interpolation occurs. See the discussion for the Interpolate (Parameter) transform operator.

InterpolateRow(s) Returns the value of s at the current pixel if it is visible and the interpolated value if it is invisible. The interpolated value is computed as the value of the straight line segment between the nearest visible pixels in the same row of s. If all pixels in the current row of s are invisible, the function returns Null. Overall, this function works much faster than Interpolate but is also less accurate.

Int(x) Returns the largest integer that is less than or equal to x (same as Floor(x)).

Kriging(drawing, column, neighbors, model, voronoi) Interpolates objects in the given drawing using the given column as the height. Line objects participate as sets of their coordinates. Area objects are ignored. Setting the number of neighbors to -1 will use the default number of neighbors. The model argument is a case-insensitive string, e.g., "spherical" or "gaussian". The voronoi argument is a boolean (if true, only Voronoi neighbors are used). All arguments except the first two are optional.

KrigingMedianPolish(drawing, column, neighbors, model, voronoi) Interpolates objects in the given drawing using the given column as the height using the median-polish kriging interpolation method. Line objects participate as sets of their coordinates. Area objects are ignored. Setting the number of neighbors to -1 will use the default number of neighbors. The model argument is a case-insensitive string, e.g., "spherical" or "gaussian". The voronoi argument is a boolean (if true, only Voronoi neighbors are used). All arguments except the first two are
optional.

**Laplace1(s)** Returns the value of the Laplace1 filter. CUDA enabled.

**Laplace2(s)** Returns the value of the Laplace2 filter. CUDA enabled.

**Latitude()** Returns the latitude of the current pixel in standard lat/lon coordinate system (lat/lon, WGS84).

**Log(x)** Returns the natural logarithm of x.

**Log2(x)** Returns the binary logarithm of x.

**Log10(x)** Returns the decimal logarithm of x.

**Longitude()** Returns the longitude of the current pixel in standard lat/lon coordinate system (lat/lon, WGS84).

**LowPass1(s)** Returns the value of the LowPass1 filter. CUDA enabled.

**LowPass2(s)** Returns the value of the LowPass2 filter. CUDA enabled.

**LowPass3(s)** Returns the value of the LowPass3 filter. CUDA enabled.

**MajValue(s, w)** Returns the most frequently occurring value of s in a window of specified size (1 for 3x3, 2 for 5x5, etc). In case of a tie, returns the lowest of the most frequently occurring values. CUDA enabled. With CUDA, the maximum window size is 31.

**Max(x, y)** Returns the maximum value of x and y.

**MaxValue(s, w)** Returns the maximum value of s in a window of specified size (1 for 3x3, 2 for 5x5, etc) or the maximum value of s overall. CUDA enabled. With CUDA, the maximum window size is 31.

**MedianCross(s)** Returns the value of the MedianCross filter. CUDA enabled.

**MedianSquare(s)** Returns the value of the MedianSquare filter. CUDA enabled.

**MedianSquare5(s)** Returns the value of the MedianSquare5 filter. CUDA enabled.

**MedValue(s, w)** Returns the median value of s in a window of specified size (1 for 3x3, 2 for 5x5, etc). CUDA enabled. With CUDA, the maximum window size is 31.

**Min(x, y)** Returns the minimum value of x and y.

**MinValue(s, w)** Returns the minimum value of s in a window of specified size (1 for 3x3, 2 for 5x5, etc) or the minimum value of s overall. CUDA enabled. With CUDA, the maximum window size is 31.

**Mod(x, y)** Returns the remainder of the division of x by y. y must not be equal to 0. Examples: (Mod(5.2, 1.0) = 0.2; Mod(5.2, 0.3) = 0.1).

**PixelArea(unit)** Returns the area of a single pixel in the context surface in the given units.

**PixelHeight(unit)** Returns the height of a single pixel in the context surface in the given units.

**PixelWidth(unit)** Returns the width of a single pixel in the context surface in the given units.

**Pow (x, y)** Returns the value of x raised to the power of y.
Surfaces and Terrains

Rad2Deg(x)  Converts x from radians to degrees.
Rnd()       Returns random value between 0 and 1.
Sgn(x)      Returns the sign of x (-1 if x is negative, 1 if x is positive, 0 if x is 0).
Sharpen(s)  Returns the value of the Sharpen filter. CUDA enabled.
SharpenMore(s) Returns the value of the SharpenMore filter. CUDA enabled.
Sin(x)      Returns the sine of x.
Slope(s, w) Returns the slope of s in the range of 0 to 90 degrees, computed over a window of specified size (1 for 3x3, 2 for 5x5, etc). CUDA enabled.
Sqr(x)      Returns the square root of x.
SumValue(s, w) Returns the summary value of s in a window of specified size (1 for 3x3, 2 for 5x5, etc). CUDA enabled. With CUDA, the maximum window size is 31.
Tan(x)      Returns the tangent of x.
Tile(s,x)   Returns the value of the Tile filter for surface s and tile size x pixels. CUDA enabled.
TileMedian(s,x) Returns the value of the TileMedian filter for surface s and tile size x pixels. CUDA enabled.
Triangulation(drawing, column, corners) Interpolates objects in the given drawing using the given column as the height using the triangulation interpolation method. Line objects participate as sets of their coordinates. Area objects are ignored. The corners argument is the height to use for corners and can be omitted.
TriangulationContours(drawing, column, corners) Interpolates objects in the given drawing using the given column as the height using the contour triangulation interpolation method. Line objects participate as sets of their segments, with intersections between segments being resolved automatically. Area objects are ignored. The corners argument is the height to use for corners and can be omitted.
Value(s, dx, dy) Returns the value of surface s at current position offset by dx and dy.
Width(s)    Returns the number of columns in surface s.
X()         Returns the X coordinate of the current pixel in coordinate system of the target surface.
Y()         Returns the Y coordinate of the current pixel in coordinate system of the target surface.

s as an argument indicates the function takes the name of a surface (within square brackets []) if the name has a space character in it) as an argument. x and y as arguments are numeric arguments. CUDA enabled operators will execute automatically within NVIDIA CUDA multiprocessor hardware, if available and enabled. See the NVIDIA CUDA topic.

The filter functions in the above are analogous to the convolution matrix filters used in image transformation via the image transform toolbar. See the Transform Operators - Images for a short description of each filter.

Operators
In Boolean operators, a value of 0 is considered to be False, and all other values are considered to be True. All Boolean operators return either 0 or 1.

The following Boolean operators are available:

- **x And y**: Returns True (1) if both x and y are True, returns False (0) otherwise.
- **x Eqv y**: Returns True (1) if both x and y are True or False, returns False (0) otherwise.
- **x Imp y**: Returns True (1) unless x is True and y is False, returns False (0) if x is True and y is False.
- **Not x**: Returns True (1) if x is False, returns False (0) otherwise.
- **x Or y**: Returns True (1) if either of x and y is True, returns False (0) otherwise.
- **x Xor y**: Returns True (1) if one of x and y is True and another is False, returns False (0) otherwise.

**Constants**

Pi (3.1415926...) and Null are the only available constants. Null is used to create invisible pixels. See the Transforming Surfaces topic for examples.

**See Also**

NVIDIA CUDA
Surface Tools
Transforming Surfaces
Visible Area

The Drawing - Visible Area menu selection is enabled when the optional Surface Tools extension has been installed and the focus is on a drawing layer in a map window that also includes a surface layer. If you have not activated the Surface Tools extension with a valid Surface Tools serial number you will not be able to use the Drawing - Visible Area command.

The Visible Area command works with a surface and a drawing to find all parts of the surface that are visible from one or more locations in the drawing marked by points. This function in other GIS systems may be called a viewshed function or visibility zones function. This function is bi-directional in that it may also be used to find all locations from which the tops of one or more towers are visible from the surface.

The Visible Area command creates a new area object within the target drawing that shows the region of the surface that is visible from the designated points.

Launching the Visible Area command in the Drawing menu launches the Visible Area dialog. The dialog above uses a surface called Montara Mountain and a target drawing called Drawing (not a very original name, but descriptive).

Controls

- **Source**: Choose the surface for which visibility is to be analyzed.
- **Target**: Choose the drawing from which viewing point locations will be taken and within which the visible area will be created.
- **Scope**: [All Objects in drawingname] or the selection or a saved selection, if one exists.
- **Height**: The data field to be used for heights of observing points. Choose [None] if no height field is to be used, in which case the observing points will be taken to lie upon the surface.
- **Relative Heights**: If unchecked, interpret the height field to mean an absolute altitude. If checked, interpret the height field to mean a relative height above the surface at that location.
- **Compute area visible from all locations**: If unchecked and multiple points are given the created area will be the region visible from any one of the locations. If checked, the area will represent only that part of the region that is simultaneously visible from all of the locations.

Using Heights

If [None] is specified in the Height field, Manifold will compute the regions of the surface visible from each location assuming that the observer's eye is located at ground level. That usually is not a realistic choice since a
literally ground level view will easily be obstructed by slight rises that would not impede the view even from the height of an average person's eye.

It is much more frequently the case that we would like to know areas that are visible from a person's eye or from a tower or other structure. In such cases the eye of the observer is elevated to some height above the surface. The **Height** option allows us to specify the name of a column in the drawing's table that provides the heights of each viewing location.

At times the height of the viewing location is known in absolute terms (such as the height of an aircraft observer) and at times the height of the viewing location is known in relative terms (such as the height of a tower above ground level). The **Relative heights** option box allows us to choose which is the case for the height values in our drawing's table. Check this option box to use relative heights, for example, the height of a tower above ground level.

**Example**

This example computes the area visible from five towers in the region of Montara Mountain, California. It uses a drawing of five points and a surface taken from part of the Montara Mountain USGS SDTS DEM data set.

Imported into the project the components appear in the project pane as seen above. The drawing (called **Drawing**) and the surface (called **Montara Mountain**) appear together in a map (called **Map**).

Opening the map we see that the drawing contains five points.
Opening the drawing's table, we see that each point has a **Height** value ranging from 100 to 10. The points have been thematically formatted in the drawing so that the color of each point varies from yellow (the 10 point) to purple (the two 100 points located on ridges).

The drawing represents five towers ranging in height from 10 feet to 100 feet that are located in various parts of the Montara Mountain region.

To find the visible area we click on the drawing tab in the map and launch **Drawing - Visible Area**.

In the **Visible Area** dialog we choose **Montara Mountain** as the Source, **Drawing** as the Target, [All Objects in Drawing] as the Scope and **Height** for the Height field. We check the **Relative heights** box because the height for each tower is given as the height above ground level at that position. Press **OK**.
The result is that a new area is added to the drawing. We have taken a moment to format the area in yellow color so that it is easy to see. Any location on the surface within that area is within sight of the top of one of the five towers. Because there are five towers and three of the towers are high towers, much of the surface is visible from at least one of the towers.

Suppose we would like to determine the visible area from only a subset of towers? That’s easy to accomplish.

Let’s suppose we are interested only in the area that is visible from either of the two tallest towers. We delete the new area created in the previous step, and then we select the two tallest towers (the purple dots).
We launch the Visible Areas dialog as before except this time the Scope is set to [Selection in Drawing]. Press OK.

The result is a new area that shows all parts of the surface that are visible from one of the two selected towers. Although it is a smaller area than is visible from all five towers, nonetheless it covers a large part of the surface because the two selected towers are tall towers that are located on high ridges.

So far we have looked at two cases where the visible area is visible from any one of the towers. Suppose we would like to find those regions that are visible from both towers at once?
We could delete the visible area created in the previous step, select the two tall towers and then re-run the Visible Areas dialog. This time we can check the Compute area visible from all locations box.

The result is a new area that shows all parts of the surface visible to both of the two selected towers at the same time. Because line of sight is bi-directional, if we stood at any of the locations marked by the yellow area we would be able to see the tops of both towers.

If we like, we can see which area is visible from all five towers simultaneously.
To do so we launch the **Visible Area** dialog using **[All Objects in Drawing]** for the **Scope** and we check the **Compute area visible from all locations** box.

The result shows that very few locations can be seen from all five towers at the same time. In general, it is only portions of the highest ridges that may be seen from all five towers at once. If we stood at any location within the yellow area we could see all five towers at once.

**Notes**

Only one area object is created by the **Visible Area** command. The area is a branched area object so that even those parts of the area that appear disconnected from other parts of the area all belong to the same area object. Clicking on any one part of the area using touch selection will select the entire area.

Strictly speaking, if we want to consider the visibility of a 100-foot tower we should add some value for the height of the individual who is viewing. One way to do this is to add some average amount, say, 5.5 feet, to the height of each tower.

Line of sight is bi-directional. Every spot that may be seen from the top of a tower is also a spot from which the top of a tower may be seen. The **Visible Area** command not only lets us find all locations that are visible from a given location, it also allows us to find all areas from which the given location is visible. This function is very
useful in determining whether or not towers or other structures can be seen. For example, if we would like to
determine the areas from which a cellular telephone antenna may be seen we can determine the visible area from
the cellular antenna's location using the height of the antenna. Anyone standing within the visible area will be
able to see the antenna.

This command is very useful for planning any line of sight task. For example, it can be invaluable when planning
the location of monitoring instruments (such as seismic sensors on volcanoes) that connect to central stations via
microwave or laser communications links, the location of observation posts for borders or secure installations or
even the location of defensive positions in military installations.

Computing visible areas can be a highly computationally intensive task that can take a long time to accomplish
with large surfaces or with many viewing points.

**See Also**

**Surface Tools**
Profiles and Elevations

Profiles and Elevations are supplementary components that are available within a Manifold project when the optional Surface Tools extension has been installed. If you have not activated the Surface Tools extension with a valid serial number you will not be able to use Profiles or Elevations.

A profile is a Manifold component created by copying a line from a drawing and then pasting it as a profile, choosing a surface to which that profile will be bound. The line in the profile shows a path over the surface. By themselves, profiles have little utility except that they are used to create elevations.

An elevation is a Manifold component that is a chart showing the height of a surface plotted along the length of a profile path. The appearance of an elevation is controlled by opening the elevation and then choosing View - Display Options. There can be more than one elevation bound to the same profile. For example, we can right click on an elevation in the project pane, choose Duplicate and then open that new elevation and change the Display Options for that elevation. This would provide two elevations bound to the same profile but showing the results in different ways.

Profiles are normally created by showing a surface in a map together with one or more drawing layers. A line for a profile may be chosen by selecting an existing line in a drawing, or by creating a new line in a drawing. Because Manifold already has a wide range of tools for creating and editing lines in drawings, we can use a drawing to create whatever line we want for the profile.

Once the line is created, we can copy it and then paste it as a profile. When a profile is created, it appears in the project pane beneath its parent surface. An elevation is automatically created under its parent profile as well.

Profiles may participate within maps as layers, they may be re-projected, and they can appear within print layouts. Profiles may have control points and may be georegistered, and they support use of the layers and views panes. Profiles may even participate in web pages via the IMS File - Export Web Page command. Grids and graticules may be used within a profile. Elevations may appear within print layouts as well. Profiles and Elevations may be immediately printed using File – Print.

Profiles and elevations may be copied and pasted within the project pane. Copying a profile and pasting it will bind it to the same surface. Copying and pasting an elevation will bind it to the same profile. We can copy and paste drawings to profiles and vice versa. When copying a drawing and pasting it as a profile, if the drawing contains no lines the paste operation will fail. If the drawing contains more than one line, the profile will be created from the first line encountered within the drawing's internal ID list of lines. If the first line is a multi-branched object containing more than one branch the profile will be created from the first branch.

Surfaces and their profiles and elevations are dynamically updated. Changes made in the surface will automatically appear in the surface's profiles and elevations. Changes made in an elevation (for example, by relocating or changing the shape of the profile line) will automatically appear in the profile's elevations.

Profiles may be navigated using the standard navigation toolbar to zoom, center, pan with the grabber or measure distances with the tracker tool. Profiles may also be navigated using hot scroll. Elevations present a fixed view of the elevation chart that is always scaled to fit the available window.

Example

Suppose we are working with a surface called Montara Mountain, using the example Montara Mountain 1:24K-scale SDTS DEM surface on the Manifold CD.
We can create a map that uses this surface as one layer and overlays upon it a drawing layer showing roads in the region.

If we are interested in the changes in elevation along a given road, we can select one of the road lines as seen above and then create a profile. To create a profile, we Copy the line and then right click into the project pane and Paste as a profile.

The Paste Profile dialog allows us to choose which surface will be the parent for the resulting profile.

We press OK and see that a new profile is added to the project as a component under the parent Montara Mountain surface. Under the new profile is an elevation component.
If we click open the new profile component we see that it shows the line that was pasted as a profile, with all coordinates defining the line marked. The surface is shown in the background with partial transparency to provide visual orientation. The last coordinate in the profile is displayed using a solid black diamond icon to show the direction of the profile line.

If we click open the new elevation component we can see that it shows a chart of elevations along the path of the profile. The elevation chart reads from left to right from the beginning of the profile line to the end of the profile line. The vertical axis labels use even breaks of altitude. The elevation line will be dotted where it crosses any regions of invisible pixels in the underlying surface.

If we prefer to see the profile with full color background, we can drag and drop the profile into the map as a layer. When profiles are used as a layer within maps their faint background is not used and just the profile line (with coordinate handles) appears.
**View - Display Options**

We can change labels and other options in the elevation using the **View - Display Options** dialog.

For example, the display above has the **Label Surface Breaks** option checked so that height axis labels are automatically taken from the interval breakpoints that define the palette used to color the parent surface. The above illustration also has the **Include zero height** option checked so that the full range of altitudes is visible and not just those traversed by the profile.

Profiles also have a **View - Display Options** dialog, which is used to:

- Specify whether the coordinates defining the profile line are displayed as small edit handles.
- Set the number of decimal digits used when reporting the average, minimum and maximum heights for the profile in the **info** pane.

### Display Options for Elevations

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Shade background</strong></td>
<td>Color the background of the elevation using the same palette (if any) used to color the parent surface. On by default.</td>
</tr>
<tr>
<td><strong>Display coordinates</strong></td>
<td>Display dots at the location of each coordinate that defines the line used as a profile.</td>
</tr>
</tbody>
</table>

#### X axis (distances)

- **Label coordinates**
  - Draw vertical lines and label the X axis at each coordinate that defines the profile line. Off by default.
- **Label even values**
  - Draw vertical lines and label the X axis at even values of distance. On by default.
- **Decimal digits**
  - The number of digits past the decimal point to show in labels on the X axis. The default value of **0** results in no digits past the decimal point.

#### Y axis (heights)

- **Include zero height**
  - Show the entire range of elevations for the surface, from zero to the highest in the elevation display. Off by default, so that the full range of the elevation display is used to show the heights within the elevation only.
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label coordinates</td>
<td>Draw horizontal lines and label the Y axis at each coordinate that defines the profile line. Off by default.</td>
</tr>
<tr>
<td>Label minimum / maximum</td>
<td>Draw horizontal lines and label the Y axis at the minimum and maximum values of height that occur in the profile line. Off by default.</td>
</tr>
<tr>
<td>Label surface breaks</td>
<td>Draw horizontal lines and label the Y axis at the numeric values that define interval breaks in the palette used to color the surface. Off by default.</td>
</tr>
<tr>
<td>Label even values</td>
<td>Draw horizontal lines and label the Y axis at even values of height. On by default.</td>
</tr>
<tr>
<td>Decimal digits</td>
<td>The number of digits past the decimal point to show in labels on the Y axis. The default value of 0 results in no digits past the decimal point.</td>
</tr>
</tbody>
</table>

Ok: Accept the current display option settings and update the Elevation display.

Cancel: Exit the dialog without making any changes.

**Editing Profiles**

The shape of profiles may be changed by **CTRL-ALT** clicking the profile line within the profile and then dragging the coordinate edit handles to new locations. **CTRL**-dragging an edit handle will move the entire profile line. This is a handy way of seeing the elevation along different paths if the profile's elevation window is kept open while dragging profile edit handles around to new locations: as soon as the edit handle is moved, the elevation window will show the new elevation automatically.

For example, suppose we start with a profile like that seen above, a straight line defined by just two coordinates. Note that the ending coordinate is shown with a solid black diamond icon.
The elevation for that profile shows the heights traversed by the profile line. The ending coordinate is shown with a solid black diamond icon so that we may easily relate the direction of the elevation to the direction of the profile line. Note the flat sections where the profile line crosses over Pilarcitos Lake.

To select the profile line for editing we **CTRL-ALT** click on it.

Edit handles will appear at each coordinate defining the line. We can then click and drag an edit handle to move it.

Dragging the edit handle automatically reshapes the line.

When we release the drag at a new position the elevation will automatically be recomputed.
If we keep the elevation window open while editing the profile we can immediately see the different altitudes traversed by the new profile line.

Profile lines may also be edited by CTRL-ALT clicking the line to select it for editing and then right clicking onto the line and choosing a function from the Coordinate context menu choice. Coordinate - Add, Coordinate - Delete and Coordinate - Duplicate commands may be used to add, delete and duplicate coordinates (and their editing handles) which may then be dragged to new locations as desired. Note that as edits are made any elevation window for that profile that is open will be dynamically updated. For examples of editing using the Coordinate context menu, see the Changing the Shape of Objects section in the Editing Objects topic.

Edits to profiles support single step Undo and Redo. That is the editing command just performed may be undone or redone.

**Invisible Pixels**

Where a profile crosses a region of invisible pixels in the surface the profile line will be shown in the elevation using a dotted style.

For example, suppose some pixels were deleted from the Montara surface as seen in the profile above.
In the elevation for that profile, the elevation line will be drawn using a dotted line style for the portion of the profile that crosses the region of invisible pixels.

**Saving Elevations as Images**

The Tools - Make Image command captures the current scene displayed in an elevation window as an image. Images made of elevation windows are captured at screen resolution.

**Info Pane and Profiles and Elevations**

When the focus is on a profile or elevation window or on a profile layer in a map, the info pane will report the average height for the profile as well as the minimum and maximum height in the profile.

**Use in Layouts**

Profiles and elevations may be used as elements in print layouts just like other components, either as their own elements or as a layer in a map that appears as an element in the print layout. When an elevation element is used in a print layout, if it is selected for editing the font used may be changed using the layout's format toolbar.

Text entries in a print layout that are bound to profiles and elevations can use [**Average Height**], [**Minimum Height**], and [**Maximum Height**] escape sequences to automatically incorporate these numbers into the text.

**Example**

We can create a print layout using an elevation, or we can drop an elevation element into an existing layout.

In the illustration above the page setup uses landscape orientation and the elevation element has been resized and moved to the left of the page.
We can insert text into the layout using the **Insert Text** button. Click on the **Insert Text** button and then draw a text box in which the text will be located.

In the **Insert Text** dialog, enter the text that is desired. We have used the **[Average Height]**, **[Maximum Height]**, and **[Minimum Height]** system generated expressions to add these values to the text automatically. See the **How to Print** topic for a detailed example of adding text to a layout.
The new text appears in the layout.

If we zoom into the layout we can see how the numbers for average, maximum and minimum height are automatically generated and substituted into the text element. Note that if we change the profile line or surface, the resultant elevation will also change and the reported numbers for average, maximum and minimum values will also automatically change.

**Note**

The foreground and background colors of profiles can be set in **Tools - Options - Colors**.

Although we may edit the profile line in simple ways within the profile, for more complex editing we can use the full roster of editing tools within drawing windows to create and edit lines as desired to use for profiles.
Analysis

Manifold provides a wealth of analytic capabilities. These include a full range of tools from simple measurement to complex statistics or network-analytic measurements. Analytic capabilities are distributed throughout the system where they are needed during various tasks. Highlights include:

**Transform Toolbar**
Transform toolbar operators provide numerous analytic functions. In drawings, these include spatial analytic functions such as the creation of Buffer Zones or Voronoi tilings or many other functions. The transform toolbar operators for images and tables likewise provide many analytic functions, including numerous statistical functions for tables.

**Decision Support System**
The Decision Support System allows analysis through queries using flexible criteria. It also provides the fuzzy inference engine used to power the "More like this" analytic capabilities built into Manifold tables.

**Active Columns**
Use formulas within active columns to add automatically computed measurements to tables. For example, we can create an active column titled "Perimeter" that contains a simple formula that reports the perimeter of the object for each record.

**ViewBots**
ViewBots provide a one-line analysis bar that is aimed at a window and which reports some analysis based on the contents of that window. ViewBots are used to perform simple statistics, queries, and other functions, including measurement functions such as the total area of selected objects.

**Queries**
Manifold SQL provides many analytic functions, including measurements, spatial relationships, and statistics.

**Spatial Overlay**
Spatial overlays provide a way of combining data between areas, lines and points that is often used in analyses.

**Surfaces Display Options**
Display options for surfaces can provide Height, Slope and Aspect analyses.

**Tracker**
Use the Tracker tool to interactively measure distances and areas by clicking with the mouse.

**Scripts**
Scripts may be written within Manifold to provide any analytic function.

In addition to the above major areas, there are numerous "implied" analytic capabilities throughout the system. For example, the Threshold function with images provides an embedded analytic capability that creates a histogram of pixel values. This histogram can then be used to set ranges for forcing pixels to white or black values.

Many types of implied analysis appear in the presentation capabilities of Manifold as well. For example, when creating a thematic format we have many choices as to how data fields will control the formatting. Some of these choices involve analysis, such as a natural breaks grouping of intervals used for thematic coloring.
Measurement

Measuring distances, areas and other quantities in Manifold is easy. There are several ways to make measurements in Manifold System:

**Tracker** Use the Tracker tool to interactively measure distances and areas by clicking with the mouse.

**Active Columns** Use formulas within active columns to add automatically computed measurements to tables.

**ViewBots** ViewBots provide a one-line analysis bar that is aimed at a window and which reports some analysis based on the contents of that window. ViewBots are used to perform simple statistics, queries, and other functions, including measurement functions such as the total area of selected objects.

**Queries** Manifold SQL provides many measurement functions.

**Scripts** Scripts may be written within Manifold that provide custom measurement functions.
Tracker

The tracker tool provides interactive distance and area measurements. To use the tracker, click on any start location. As the mouse is moved, the status bar will provide a distance from the starting point to the current position of the cursor.

If we click the tracker at several locations the distance reported will be through the polyline distance from the starting point to the current cursor location. Right click or press the ESC key to clear the tracker so it may be clicked at a new starting point.

For example, if we wish to know the distance from the Oakland Estuary around Treasure Island, back to the City front and then up to Sausalito we can begin by clicking the tracker cursor just outside the Estuary. We can then click a few more times to form a line going around Treasure Island and back to the City. At all times the status bar measurement readout will report the distance through the resultant line from the starting point to the current position of the cursor.

When we move the cursor up towards Sausalito we can see the distance over the indicated path is about 21671 meters.

To measure exact distances between specific locations, we will often use Snap together with the tracker tool to snap to an exact location, such as a point. Use the TAB key to jump the cursor to the nearest snap location as specified by the current Snap To modes.

By default, the tracker tool will report distances using the unit of measure used in the native coordinate system (projection) of the drawing. Pressing CTRL will shift the unit of measure to the units set in the Tools - Options dialogs. When using CTRL, distances will be computed over the Earth ellipsoid in use and reported in linear measurement units even if the drawing has not been projected.

Autoscroll

The tracker tool is a "live" cursor that is attached to the mouse cursor. Moving the cursor out of the active window after the first tracker click will cause an autoscroll. To avoid unintended autoscroll, remember to use ESC or a right click to end tracker scrolling before moving the mouse cursor out of the active window.

Press CTRL to Change Units and Computation

Press and hold down the CTRL key at any time to switch the tracker read out between native units and the default measurement unit specified in Tools - Options.

For example, in maps using Latitude / Longitude projection the native units are degrees. If we would like to see tracker distance in miles or kilometers we would hold down the CTRL key. The tracker in CTRL mode will show measurements in either English (miles and feet) or metric (kilometers and meters) depending on the setting of the Use English measurement units option in Tools - Options. Small distances will automatically use the smaller units (feet or meters) with the display jumping to larger units (miles or kilometers) for larger distances. Very small values will be reported using millimeters or inches.

Using CTRL also shifts the formulae used within Manifold to compute distance. The default tracker measurement uses plane geometry on the assumption that the component is seen in whatever projection is desired. By default, the tracker tool reports simply Euclidean distance in whatever units define the native, planar coordinate system of the drawing. Pressing CTRL shifts the computation to a full ellipsoidal trigonometric computation of the arc distance of the geodetic curves involved for length or areas.
The difference may be seen sometimes when using metric units so that the default tracker might report a distance of 50017 meters while a CTRL tracker reports, for example, 49.92 kilometers. The difference between the two is the difference between an Euclidean computation on a plane and an ellipsoidal computation on the Earth ellipsoid in use.

**Accuracy**

Normally, if a projection is well chosen for the region of interest the difference between the two computations (using or not using CTRL) will not be that great. A large difference between the two computations indicates either an error in georegistration or the inappropriate use of a projection. For example, GIS newbies sometimes attempt to combine data from multiple UTM zones into a single UTM zone, applying the single UTM zone to geographic regions far beyond where that particular zone's projection will be accurate. In such cases, there will be a significant difference between the value reported in native units as a result of a planar Euclidean computation and the value reported in default measurement units as result of an ellipsoidal trigonometric computation.

For highest accuracy first make sure that the component is correctly georegistered and second make sure that the projection used for the image is one that is scale-invariant and centered exactly upon the region of interest.

Because no projection is perfect at representing the curved surface of the globe, when measuring very large regions of the Earth there is likely to be a significant difference between the ellipsoidal trigonometric computation done with CTRL and the Euclidean computation done by default, even if the projection is well chosen. In such cases the CTRL computation is the more accurate one.

At times the difference between Euclidean and ellipsoidal calculations will be great even over relatively small regions. For example, when using the tracker tool with Mercator projections (which do not preserve distances accurately away from the Equator) there will be great differences between the tracker and the CTRL-tracker measurement. This effect may be very noticeable when using image servers to provide linked images in Mercator projection. When making measurements in such cases use the CTRL-tracker numbers.

**Measuring Perimeter and Area with SHIFT**

The tracker may also be used to measure area. The tracker reports the area within the closed region formed by connecting the starting point to the current position of the mouse. Press the SHIFT key at any time when using the tracker and the area denoted by the tracker path so far will be closed and a computed area will be reported to the status bar.

The SHIFT and CTRL keys may be combined to show areas in different units.

**Example**

The illustrations below show a drawing with three points. The drawing is in unprojected Latitude / Longitude so the native units used are degrees. We have set Snap To Points and have clicked on the tracker.

We first click on the uppermost point. As the cursor moves near the middle point the snap function will snap the square secondary cursor onto that point and…

![Image](image-url)

... the status bar will report the length of that segment in kilometers. At the scale used kilometers are a better match than meters. We can change the default units between metric and English (miles) units in Tools - Options.

Pressing the CTRL key while operating the tracker will toggle the units shown between native units (degrees, in this case) and "real" units (miles or kilometers).
We can click to anchor that tracker segment on the middle point. Now, when we move the tracker cursor near the third point the *snap* function will snap / extend the tracker line to the third point.

With the square cursor box on the third point the status bar will read the overall length of the tracker line, which now consists of two segments.

At any time we can press the *SHIFT* key to temporarily close the tracker line back to the originating point. In the above illustration, we’ve positioned the mouse cursor near the third point so that the tracker line snaps to the third point, but we have also pressed the *SHIFT* key so that the tracker line closes the figure back to the first point.

The result in the status bar is the length of the entire tracker perimeter as well as the area in square kilometers.

**Scroll Wheel**

When using a mouse with a scroll wheel we can use the scroll wheel to zoom in and out. This works even in the middle of commands such as the tracker tool. By default the zoom will be centered on the mouse position. Holding the *CTRL* key while operating the scroll wheel will force the zoom to be near the center of the opened window.

**Notes**

Keep in mind that projected maps use linear measures such as feet or meters as their native units. Toggling the tracker tool using *CTRL* may therefore in some cases toggle between the native unit and "real" units that in both cases may be meters.

In addition to using a right click or the *Esc* key to cancel tracker function, the *ALT-Tab* key combination may also be pressed if some non-Manifold program or information dialog (such as a "download complete" notification when downloading files in background) has snatched context away from the Manifold window.

When used with layouts, the tracker tool will report in either English (inches) or metric units depending on the setting of the *Use English measurement units* option in *Tools - Options*.

Tracker computations depend upon correct georegistration of the component being measured. If the component has not been correctly imported and georegistered the *CTRL* computation cannot be accurately done.

**Advanced**

See the Dialog Mode and Visual Tools topic for a parameter or value oriented way of specifying a tracker path or shape.
ViewBots

ViewBots are one-line analytic instruments used to dynamically compute a statistical or comparative measure over a subset of records and are extremely useful for analyzing tables. They are one of the most popular functions in Manifold for experienced users. When a ViewBot is configured for a drawing's table it continues to function when the drawing has the focus. For example, if we configure a ViewBot to show the average of a given field for selected items it will continue reporting the average of the selection whether or not we select items from a drawing's table or by using mouse selection commands to select objects in the drawing.

Turn on the ViewBots pane with a \texttt{SHIFT-ALT-B} ("B" for "ViewBot") or with a \texttt{View - Panes - ViewBots} command.

The ViewBots above, for example, report on those operations they have been told to conduct on the table in view. ViewBots are bound to the tables for which they were created. When a drawing window is active, the ViewBots pane will list all ViewBots created for the drawing's table. When a drawing layer in a map is active, the ViewBots pane will list all ViewBots for the active drawing's table.

To Create a ViewBot

1. Open the ViewBots pane and press the \texttt{New ViewBot} button. A new ViewBot line will be added to the ViewBots pane.
2. Double click onto the saved ViewBot fields to choose analytic functions and the scope of the ViewBot.
3. Press the \texttt{Refresh} button to update the results in that ViewBot if the ViewBot autorefresh option is not on in \texttt{Tools - Options}.

To Make a Selection using a ViewBot

1. Click on the target component to make it active. Open the ViewBots pane.
2. Click on the desired ViewBot to highlight it.
3. Click on one of the ViewBot pane's selection command buttons to combine the object set picked out by the ViewBot with any selection that is currently made. For example, clicking on the \texttt{Subtract from Selection} button will take the set of objects picked out by the ViewBot and will subtract the objects it contains from the current selection.

To Delete a ViewBot

1. Click on the desired saved ViewBot to highlight it.
2. Press the \texttt{Delete ViewBot} button.

To Edit a ViewBot

1. Double-click on the ViewBot to be edited, or
2. Click on the ViewBot to highlight it and press the \texttt{F2} key.
ViewBots Pane Toolbar Commands

Use the New ViewBot and Delete ViewBot buttons to create and to remove ViewBots from the pane. Use Refresh and Refresh All to update the calculation in the highlighted ViewBot or in all ViewBots.

Five selection command buttons are arrayed at the top of the ViewBots pane. These commands apply to whatever set of objects is picked out by the highlighted ViewBot. If we click on a ViewBot to highlight it and then press one of these buttons, the object set picked out by that ViewBot will be combined with whatever is the current selection in the image. The selection commands allow us to use object picked out by a View ViewBot to replace the selection, to add to the selection, to subtract from the selection, to invert with the selection and to intersect the selection. For example, if we have a ViewBot that reports the number of objects for which the Elevation is greater than 100, the set of all objects picked out by this ViewBot will be combined with the existing selection with the selection command buttons.

- **Move Up** - Move the active layer up one position in the layer stack.
- **Move Down** - Move the active layer down one position in the layer stack.
- **Replace Selection** - Replace the existing selection with the objects picked out by the highlighted ViewBot.
- **Add to Selection** - Add the objects picked out by the highlighted ViewBot to the existing selection.
- **Subtract from Selection** - Subtract the objects picked out by the highlighted ViewBot from the existing selection.
- **Invert with Selection** - Deselect what is in the objects set picked out by the highlighted ViewBot that was already in the existing selection and otherwise add the ViewBot's object set.
- **Intersect Selection** - Select only the region of overlap between the highlighted ViewBot's objects and what is in the existing selection. If there is no overlap, nothing will be selected.
- **Refresh** - Update the highlighted ViewBot.
- **Refresh All** - Update all ViewBots in the pane.
- **New ViewBot** - Create a new ViewBot line in the ViewBots list.
- **Delete ViewBot** - Delete the highlighted ViewBot.

Add ViewBot Dialog Controls

- **Scope** - The objects upon which the ViewBot operates: all objects (or records, pixels, etc), the selection, or the name of any saved selection.
- **Column** - The name of the field to use. Optional for some operations. For example, Number of Areas or Center X don't require a field.
- **Operation** - The ViewBot operator to be employed.
- **Argument** - Arguments, if required for the operator.
- **Ignore Case** - Treat upper and lower case letters the same in strings.
Used with text operators.

**Ignore leading and trailing whitespace**
Ignore any space or tab characters in strings. Used with text operators. "This" would be treated the same as "This".

**Ignore interior whitespace**
Ignore any space or tab characters within strings. Used with text operators. "Th is" would be treated the same as "This".

**Caption**
String to display for ViewBot results. Can include escape sequences in square brackets:

- **[Argument]** - Replace with argument in use.
- **[Column]** - Data field in use.
- **[Operation]** - Replace with the operation in use.
- **[Scope]** - Replace with scope in use, such as the name of a saved selection.
- **[Scope Size]** - Replace with the number of items in the scope.
- **[Time]** - Estimated time to compute the ViewBot. Normally used when **Autorefresh** is turned off to warn users of computationally intensive ViewBots that should not be casually refreshed.
- **[Value]** - Repeats the value computed and displayed in the **Value** column. Normally redundant.

The default **Caption** string of **[Column] [Operation] [Argument] in [Scope]** creates a reasonably comprehensible sentence for most ViewBots.

**Autorefresh viewbot**
Command ViewBot to self-refresh on any changes in the subject component. If this value is not checked, a **Refresh** or **Refresh All** command in the ViewBots pane will be required to update the value of the ViewBot on any changes.

**Selection and ViewBots**

Some ViewBots such as the maximum value of a column allow selection from the ViewBot and some (such as the average value of a column) do not.

**Examples**

We will open the **Order Details** table from the sample **Nwind.mdb** database.
For simplicity, we will hide all columns except Order ID and Unit Price. In the illustration above we’ve opened the ViewBots pane using View - Panes - ViewBots and have positioned it over the table so we can make screenshots of both at the same time.

To create a new ViewBot we press New ViewBot in the ViewBots pane.

The Add ViewBot dialog opens up. We will choose Unit Price for the Column and Average for the Operation, using defaults for all other choices.

In particular, we will leave the caption string to the default settings. We press OK and a new ViewBot is created.

We can press Refresh All at any time to re-compute the ViewBot if Autorefresh viewbot is not checked. The Value column for the ViewBot reports that 24.71998… is the average of unit price values in the table.
To see how the Caption string is generated, recall that the Caption box in the Add Viewbots dialog contains the default string:

\[[Column] \[Operation] \[Argument]\] in \[Scope\]

The result in the ViewBot Caption column is the text seen above.

When the ViewBot is recomputed, the values of the various escape phrases in \[] square brackets are replaced with the relevant text. There is no \[Argument\] used in this ViewBot so it is replaced by nothing.

Let's add another ViewBot. Once again we press the New ViewBot button.

We will use the same settings as before, but this time we will change the Scope to \[Selection\]. The Scope setting specifies upon which objects (records) the ViewBot will operate. By changing the Scope to \[Selection\] we are telling the ViewBot to look only at records that are selected.
We can press **Refresh All** again (or, simply check the **Autorefresh ViewBot** box when creating the ViewBot) to see the new values. There is no value for the second ViewBot because we have not selected anything in this table.

If we make a selection as seen above and press **Refresh All** we see that the average of unit price for the selected records is **23.3**. Note that we can simultaneously see the average for all records and for those only in the selection.

Suppose we had previously made a selection representing new orders we've received and that we've saved this selection in the Selections pane under the name **New Orders**. We could create a third ViewBot using **New Orders** in the **Scope** box (all saved selections will appear as choices in the **Scope** box).

We would then be able to see how the unit price average for all records compares to the unit price average in the **New Orders** saved selection as well as in any current selection we care to make. We could also have query panes or use other methods to make selections and the ViewBots will faithfully report the latest figures based on the current selection if that is the **Scope** specified. The "live" ability of ViewBots to dynamically report values based on the current selection is an analytic and data discovery instrument of great power.

We can get rid of the third ViewBot by clicking on it to highlight it and then choosing **Delete ViewBot**.

**Using ViewBots to Make Selections**

Operations like **Average** are aggregate operations that calculate a number without identifying any specific record. Many ViewBot operations can identify specific records. We can then use the ViewBot to select those records. We will add two more ViewBots that can be used for selection.
First we add a ViewBot that finds records where the Unit Price is greater than 50.

The resulting ViewBot informs us there are 191 records that are priced higher than 50.

We will add one more ViewBot that finds records where the Unit Price is less than 50.

This ViewBot tells us the table contains 2630 records that are priced lower than 50.
To make a selection using a ViewBot, we click on it to highlight it and then we press one of the Selection Commands at the top of the ViewBots pane.

For example, we can click on the greater than 50 ViewBot and press Replace Selection.

Instantly, all those records with unit price greater than 50 are selected. In the table in view, we see only one such record (the others are scrolled out of sight). Note also that the ViewBot showing the average unit price in the selection has also been recomputed. We now know that the average unit price for products priced higher than 50 dollars is 100.86. What is the average unit price for products that are priced lower than 50 dollars?

To find out, we click on the lower than 50 ViewBot and press Replace Selection once more.
Instantly, all records with unit price less than 50 are selected. We can now see from the second ViewBot that the average price for products that sell for less than 50 dollars is 19.13.

**ViewBot Uses**

ViewBots can be used for a very wide variety of purposes. In the example above we could have added ViewBots that reported the total number of items in the selection and then tried different ViewBots at different price levels to find average, maximum and minimum values.

We can use ViewBots for selection. Suppose we have a mailing list that needs to have duplicates cleaned. We could open a ViewBot that finds all duplicates except the first. If we select records with that ViewBot we can then delete the selected records.

Suppose we have a customer database for a store where we would like to find all customers who have purchased refrigerators who are not on credit hold. If our database has a "notes" field that contains the word "refrigerator" for anyone who has purchased a refrigerator, and if it also has a field for credit hold, we can create two ViewBots. One ViewBot finds all records where the notes field contains "refrigerator." The other ViewBot finds all records where credit hold is true. To create a list for mailing, we use Replace Selection with the refrigerator ViewBot to select all records with refrigerators. We then use Subtract from Selection with the credit hold ViewBot to remove the credit hold records from the selection. The resulting selection is only those people with refrigerators who are not on credit hold.

**Example: Using ViewBots with Drawings**

ViewBots are a great way to see a report of information from a drawing. In this example we import and use the mexico_eg.mif sample drawing that is on the Manifold CD. This drawing shows provinces in Mexico and provides demographic data for each province.

Begin by importing the drawing. Open the drawing and turn on the ViewBots pane with a SHIFT-ALT-B or with a View - Panes - ViewBots menu command.

Click the New ViewBot button to create a new ViewBot.
Configure the Add ViewBot pane as seen above. In the example drawing the POBL_1990 field gives the population in 1990 for each province. We will create a ViewBot that reports the sum of population in all selected provinces. Although Manifold will offer a suggested caption based on the scope and operation in use, we will provide our own caption, “Total Population in Selection.” Note that the Autorefresh viewbot checkbox has been checked. This updates the ViewBot on any changes.

The initial result in the ViewBots pane is an empty ViewBot. The Value will not appear until we select something, either in the drawing or in the drawing's table.

If we select three provinces in the drawing the ViewBot will update.
Now it reports a total population of **5763591** for the three provinces.

If we pop open the drawing's table we can see the values of `POBL_1990` that are in the selected provinces do indeed add up to **5763591**.

If we make a different selection in the drawing the ViewBot will update the value.
We can see that this selection has a much higher total population (because, of course, it includes the provinces surrounding Mexico City).

**See Also**

See the Selection topic for examples of *Replace*, *Add*, *Subtract*, *Invert* and *Intersect* selection command usage.

ViewBot Operators for lists of operators that may be used within ViewBots.

See the Show Area of a Parcel in Acres topic for an example that uses ViewBots to show acreage of selected parcels.

ViewBot values are displayed using the format of the underlying column, if appropriate. See the Formatting Columns topic for information on setting formats.
ViewBot Operators

ViewBot operators work with fundamental Manifold data types: binary, Boolean, date, numeric and text fields. URL fields are text fields. Currency, latitude, longitude and percentage fields are numeric fields. Operator results are selectable unless otherwise noted.

ViewBot Operators for All Field Types

- **Duplicates**: The number of duplicated values.
- **Duplicates Except First**: The number of duplicated values less one for each unique value. Example: a 2-2-2-3-3-4-4 sequence of values yields 4 and a 2-2 sequence yields 1.
- **Uniques**: The number of unique values.

ViewBot Numeric, Date and Text Operators

- **Maximum**: The maximum value that occurs.
- **Minimum**: The minimum value that occurs.
- **Median**: The median of all values in the scope.
- **Less than**: The number of values less than the given value.
- **Less or Equal to**: The number of values less than or equal to the given value.
- **Equal to**: The number of values equal to the given value.
- **Not Equal to**: The number of values not equal to the given value.
- **Greater than**: The number of values greater than the given value.
- **Greater or Equal to**: The number of values greater than or equal to the given value.

ViewBot Text and URL Operators

- **Duplicates**: The number of duplicated values.
- **Duplicates Except First**: The number of duplicated values less one for each unique value. Example: a 2-2-2-3-3-4-4 sequence of values yields 4 and a 2-2 sequence yields 1.
- **Empty**: Number of empty values.
- **Uniques**: The number of unique values.
- **Maximum**: The maximum value that occurs.
- **Minimum**: The minimum value that occurs.
- **Median**: The median of all values in the scope.
- **Less than**: The number of values less than the given value.
- **Less or Equal to**: The number of values less than or equal to the given value.
- **Equal to**: The number of values equal to the given value.
- **Not Equal to**: The number of values not equal to the given value.
- **Greater than**: The number of values greater than the given value.
Greater or Equal to  The number of values greater than or equal to the given value.
Starting with The number of values starting with the given substring.
Ending with The number of values ending with the given substring.
Containing The number of values containing the given substring.
Matching The number of values matching the given regular expression pattern.
Sounding Like The number of values sounding like the given word.
Starting with Token The number of values that start with the given token.
Ending with Token The number of values that end with the given token.
Containing Token The number of values that contain the given token.
Starting with Match The number of values that start with a match to the given regular expression pattern.
Ending with Match The number of values that end with a match to the given regular expression pattern.
Containing Match The number of values that contain a match to the given regular expression pattern.

Comparison operators such as Greater than will work with text, binary, coordinate system and geometry columns as well as with numeric columns. When used with non-numeric columns a comparison operator will compare data going byte-by-byte stopping on the first different pair of bytes.

ViewBot Numeric Operators

Choose Not with some operators (such as Equal to) to create the "not" form.

Duplicates The number of duplicated values.
Duplicates Except First The number of duplicated values less one for each unique value. Example: a 2-2-2-3-3-4-4 sequence of values yields 4 and a 2-2 sequence yields 1.
Bottom Identify the lowest $N$ values where $N$ is given by the argument. Does not return a value for the ViewBot but enables selection of the lowest $N$ values.
Uniques The number of unique values.
Maximum The maximum value that occurs.
Minimum The minimum value that occurs.
Median The median of all values in the scope.
Less than The number of values less than the given value.
Less or Equal to The number of values less than or equal to the given value.
Equal to The number of values equal to the given value.
Not Equal to The number of values not equal to the given value.
Greater than The number of values greater than the given value.
Greater or Equal to The number of values greater than or equal to the given value.
### Analysis

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>The average value. Not selectable.</td>
</tr>
<tr>
<td>Sum</td>
<td>The sum of values. Not selectable.</td>
</tr>
<tr>
<td>Range</td>
<td>The range of values. Not selectable.</td>
</tr>
<tr>
<td>Moment</td>
<td>Statistical moment of the given order. Not selectable.</td>
</tr>
<tr>
<td>Center Moment</td>
<td>Center statistical moment of the given order. Not selectable.</td>
</tr>
<tr>
<td>Top</td>
<td>Identify the highest $N$ values where $N$ is given by the argument. Does not return a value for the ViewBot but enables selection of the highest $N$ values.</td>
</tr>
<tr>
<td>Typical</td>
<td>Identify the most typical $N$ values where $N$ is given by the argument. Does not return a value for the ViewBot but enables selection of the most typical $N$ values.</td>
</tr>
<tr>
<td>Variance</td>
<td>Variance of values in the scope. Not selectable.</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>Standard deviation of values in the scope. Not selectable.</td>
</tr>
<tr>
<td>Excess</td>
<td>Excess of values in the scope. Not selectable.</td>
</tr>
<tr>
<td>Skew</td>
<td>Skew of values in the scope. Not selectable.</td>
</tr>
</tbody>
</table>

**ViewBot Binary, Coordinate System and Geometry Operators**

Choose **Not** with some operators (such as **Equal to**) to create the "not" form.

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Areas</td>
<td>The number of records with geometry for area objects. Available for geometry columns.</td>
</tr>
<tr>
<td>Duplicates</td>
<td>The number of duplicated values.</td>
</tr>
<tr>
<td>Duplicates Except First</td>
<td>The number of duplicated values less one for each unique value. Example: a 2-2-2-3-3-4-4 sequence of values yields 4 and a 2-2 sequence yields 1.</td>
</tr>
<tr>
<td>Empty</td>
<td>Number of empty values.</td>
</tr>
<tr>
<td>Equal to</td>
<td>The number of values equal to the given value.</td>
</tr>
<tr>
<td>Greater or Equal to</td>
<td>The number of values greater than or equal to the given value.</td>
</tr>
<tr>
<td>Greater than</td>
<td>The number of values greater than the given value.</td>
</tr>
<tr>
<td>Less or Equal to</td>
<td>The number of values less than or equal to the given value.</td>
</tr>
<tr>
<td>Less than</td>
<td>The number of values less than the given value.</td>
</tr>
<tr>
<td>Lines</td>
<td>The number of records with geometry for line objects. Available for geometry columns.</td>
</tr>
<tr>
<td>Not Equal to</td>
<td>The number of values not equal to the given value.</td>
</tr>
<tr>
<td>Points</td>
<td>The number of records with geometry for point objects. Available for geometry columns.</td>
</tr>
<tr>
<td>Uniques</td>
<td>The number of unique values.</td>
</tr>
</tbody>
</table>

Comparison operators such as **Greater than** will work with text, binary, coordinate system and geometry columns as well as with numeric columns. When used with non-numeric columns a comparison operator will compare data going byte-by-byte stopping on the first different pair of bytes.
See Also

See Using Tokens and Text Strings for information on tokens.
Printing and Layouts

Printing

Manifold provides two paths to printing:

- The contents of any component window may be printed in a default way using the File - Print command.
- More sophisticated layouts may be composed using a Print Layout, which may then be opened and printed using File - Print. The layout will be based on whatever settings (paper size, orientation, etc.) specified in the File - Page Setup dialog. Layouts are used to print maps at a specific scale. See the Layout topic.

Note that printing items with a layout is a three-step process: create the layout, adjust the elements of the layout to taste, and then print the layout.

Page Setup

The page setup dialog allows us to set paper orientation (portrait or landscape), paper size, printer properties and borders. Page setup options will be applied to any print job.

Layouts

Layouts are used to provide more sophisticated control over printing jobs. A layout may be created using a single component or none at all. Additional components that are to appear in the layout may be dragged and dropped into the layout and then resized, moved and adjusted as desired. We can change the characteristics of components that appear within a layout by right clicking on them and choosing Properties. This allows us to change the component to use a view or to show embellishments such as a North arrow or scale bar. We can use layout tools to add text or border lines. We can then print the layout.

The most sophisticated print layouts will normally feature a main compositional element that is a map. This map will consist of many drawing, image and labels components to create the visual effect desired. Because maps provide very rich editing capabilities they are the natural "console" within which to work when creating desired effects involving many layers.

Experienced Manifold users will therefore often put most of their compositional efforts into arranging the main map as desired. They will then use that map as the main element in a layout and will add other components to the layout rather sparingly. It is often a matter of taste and convenience whether a particular compositional element is placed within a map or is dropped into the layout as a separate element.

Suppose for example we have a map of wetlands that is to be printed and we wish to have a title block that includes a main text title hovering over a corporate logo. We could create the entire ensemble with both the wetlands drawings and the text labels and image used as the title block as a set of layers in one map. Such a complex composition is shown at the end of the Layers topic using the splash screen for Manifold as an example. Once a complex composition is created as a map, it could then be dropped into a print layout that consists of just that one component.

Alternately, we could create a map showing the wetlands and then create another map that just consisted of a few labels and image layers that made up the title block. We could then drop the wetlands map into a layout and also drag and drop the title block map into the layout. We would then print the layout that now is made up of two components.

The former approach allows creating everything within the same map window so that all layers are immediately at hand to allow adjustment of any visual aspect of the map. The second approach allows us to create a standard title block that could be recycled into other print layouts. Production print jobs could well include a combination of the two methods.

Layouts also support use of layout templates which allow us to create standardized arrangements for layouts. It is very important to master the use of layout templates when engaged in repetitive layout work or when creating layouts that must share the same look and feel.

Print Toolbar Button
As is the case in standard Windows applications the Print toolbar button prints directly to the default printer. In contrast the File - Print command called from within the main menu will display a Print dialog that allows choice of which printer to use.

Printer Drivers

Printer drivers vary widely in their ability to provide rapid and efficient printing. The printer vendor normally provides printer drivers. A well-written driver will allow faster printing with less data passed between Manifold and the printer. A poorly written driver may result in slow print jobs or very large amounts of data in temporary printing files. Printer drivers with very serious defects may not print jobs correctly.

When a layout is printed, Manifold sets up the print job and passes it to Windows, which then passes the job to the printer driver. Two key options found in the printing page of the Tools - Options dialog help control how Manifold interacts with Windows and thus the printer driver.

- **Optimize printing of images and surfaces**
  - When on, the system avoids rendering image and surface components in full resolution during print job setup, deferring final rendering to the printer driver, provided the image or surface can be rendered with either no re-projection or with simple scaling and shifting. This makes the size of the print job smaller and offloads the task of scaling raster images to the printer driver. Most printer drivers can automatically re-scale images and surfaces within the driver for better performance and reduced print job size. Some drivers cannot do so and will so cause the job to be printed more slowly than if Manifold renders images and surfaces. By default the option is turned on since most modern printer drivers can automatically render and re-scale. If the print job is slow or if the printer or plotter in use is suspected to be Windows unaware, try turning this option off and see if performance improves.

- **Print using GDI+**
  - On by default. If unchecked, prints using GDI mode instead of GDI+ mode. GDI is an earlier Microsoft technology. Provided for use in case of a buggy or incomplete printer driver. Printing in GDI mode does not support antialiasing or transparency for drawings. See the Layer Opacity topic for information on transparency. Curved (rotated) or labels aligned to lines cannot be printed n Windows 9x systems if the Print using GDI+ option is turned off.

The Tools - Options dialog includes other options related to printing, but the above options have by far the largest impact on the process of printing.

If your printer driver provides poor performance or causes errors, try changing the default settings of the above two options. Make sure you have installed the most recent version of the printer driver available for your Windows operating system. Printer vendors will frequently provide new editions of printer drivers on their web sites that are more recent than the versions delivered with the printer or bundled into Windows.

It is always possible in the case of older printers or plotters that the printer vendor no longer provides driver updates, or that the only printer drivers available do not support recent Windows editions. In that case, the only choice may be to upgrade to a new printer that is supported with drivers that fully support current Windows editions.

**Tech Tip: Printing a Layout to PDF**

There are two ways to create a PDF file from a layout: exporting a layout to a PDF or "printing" the layout via File - Print using a PDF virtual printer driver, which mimics a printer to capture any printing activity as a PDF.
Printing a layout to a virtual printer driver that produces PDF feeds the layout through an ordinary GDI / GDI+ rendering pipeline, which issues "dumb" printing commands that are then captured by the driver. Exporting a layout directly as a PDF feeds the layout through the Manifold PDF rendering engine, which optimizes printing commands. In general, the results of exporting a layout as a PDF will almost always be better than the results of printing the same layout to a virtual printer driver.

Note also that printing a layout to a via File - Print exposes you to the risk of whatever that PDF printer driver vendor has in mind. For example, it could easily produce different results depending on the installed version of Acrobat if that is what is used for a PDF printer driver. In contrast, exporting a layout to a PDF via File - Export - Image will be consistently the same.

See Also

File - Print - Many components can be most easily printed by clicking open the map, drawing or image window and choosing File - Print.

How to Print - For a step by step example of creating and printing a layout.

Exporting Layouts
Layouts

Layouts are components that allow us to design printing jobs. When opened in a layout window, a layout allows us to drag and drop components such as drawings, images, maps, surfaces, tables, palettes and labels components into the layout. Charts, comments, queries and scripts can also appear in layouts. We can then resize or reposition those items to arrange the appearance of our printed page. Items in layouts may be copied and pasted within the layout. Layouts may be based upon layout templates to use standardized arrangements.

Layout windows are scaled so that the layout shown looks exactly like the printed page that will be created. Choosing View - Zoom To - 1:1 will zoom the layout to a 1:1 view that shows it on the monitor exactly as it will be printed. If we are working with US Letter sized paper in landscape orientation and we have a monitor that can show a display 11 inches by 8.5 inches, we will see a literal image of exactly how the printed page will appear.

Layouts will automatically be configured to show the printed page as it would look given current Page Setup settings for the printer selected. Layouts save their last-used printer settings when the project is saved. To change margins in the layout, use the Page Setup dialog. The Page Setup dialog uses inches or millimeters depending on the setting of the Use English Measurement Units checkbox in the Tools - Options dialog.

Layouts are not only used for printing, they may be used to create images of arbitrary DPI resolution using the Tools - Make Image command and are also used to save an image of the layout in graphics file formats such as .emf, .ps or .pdf.

Creating a Layout

Layouts may be created in two ways. The first way is to create a layout based on a particular component, such as a drawing, and to then add any other components to the layout. The second way is to create a blank layout and to then drag and drop components into the layout.

The first method will automatically scale the component that is used to create the layout to fill an entire page or, in the case of text components like tables or comments, the layout will be expanded to a multipage layout to allow the entire component to be printed. This component will automatically be the main component so that addition of any system generated text expressions will use this component. The first method is convenient when one main component will be the subject of the layout.

The second method may be more convenient when several components will be the subject of the layout, or when a layout is being created for use as a standard layout with many different projects.

To Create a Layout

1. Click on the project pane and create a new layout with File - Create - Layout. If the layout is to be based upon a particular component, choose that component in the Create Layout dialog.
2. In the Create Layout dialog, if a template is to be used check the Template box and choose a template. If the layout is to be blank, check the No parent box and no component will be loaded into the layout by default.
3. Open the new layout.
4. Verify paper size and other overall options are correctly specified in File - Page Setup.
5. Drag and drop components that are to appear from the project pane into the layout.
6. Insert any objects such as vertical or horizontal lines, text labels, etc.
7. Edit the component elements in the layout to change size, position, views to be printed, etc.
8. Use File - Print to print the layout.

Creating a layout based upon a non-text component, like a drawing, will automatically scale the component to fit the entire component on a single, full page. Creating a layout based upon a text component such as a table, query, script or comments component will automatically invoke the multipage layout capability to use as many pages as are necessary to show the entire component in the font size that is used.

Dragging and Dropping

We drag and drop items into a layout from the project pane. Begin by double clicking on the layout component to open it in a project pane. We can now click on an item in the project pane and drag and drop it into the layout window. Components that are dragged and dropped into a layout will at first appear as a fairly small layout.
element so they do not obscure other elements in the layout. Click on the component that's just been dropped to select it and then resize and reposition the element as desired.

Text Components in Layouts

Text components such as queries, scripts or comments are printed as text. Creating a layout based upon a comments, query or script component will adjust the number of pages vertically so the entire component is visible. Dragging and dropping a text component into a layout will create the usual small element that can be resized and repositioned as desired.

Text components (comments, queries and scripts) will always be printed with left alignment and will be word-wrapped to fit within the width of the layout element. Each tab will be replaced by four spaces. The default font used for the text component will be taken from the Tools - Options - Fonts page.

Selecting a text component in a layout for editing by **CTRL-ALT** clicking it will allow changing foreground color, background color and font using the format toolbar.

Inserting Items with the Tools Toolbar

The tools toolbar for layouts allows us to insert additional design elements into a layout. When a layout is the active window, the tools toolbar will change to show tools used with layouts.

<table>
<thead>
<tr>
<th>Insert Horizontal Line</th>
<th>Create a horizontal line in the layout. Default line size is 1/20th of a point.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Insert Vertical Line</td>
<td>Create a vertical line in the layout. Default line size is 1/20th of a point.</td>
</tr>
<tr>
<td>Insert Box</td>
<td>Create a rectangular box with a click and a drag. Boxes have foreground and background color and line thickness that can be set by selecting the box and using the format toolbar. Default line size is 1/20th of a point.</td>
</tr>
<tr>
<td>Insert Box on Center</td>
<td>Create a rectangular box centered on initial mouse click. Default line size is 1/20th of a point.</td>
</tr>
<tr>
<td>Insert Text</td>
<td>Add a text box, such as a caption, copyright notice or title, to the layout. Click on the text box to select it and then use the format toolbar to format the text. Double-click a text box in the layout to edit the text it contains.</td>
</tr>
<tr>
<td>Insert Legend</td>
<td>Add a legend to the layout, based upon a component used in the layout.</td>
</tr>
<tr>
<td>Insert North Arrow</td>
<td>Add a North arrow to the layout, based upon a component used in the layout.</td>
</tr>
<tr>
<td>Insert Scale Bar</td>
<td>Add a scale bar to the layout, based upon a component used in the layout.</td>
</tr>
</tbody>
</table>

When inserting horizontal or vertical lines, it is usually best to turn on the grid with View - Grid and to then use **Snap to Grid** to position layout element such as vertical and horizontal lines.

To add text, legends, North arrows and scale bars, click the tool and then click and drag open a horizontal box in the layout showing where the item is to appear. Inserting a legend, North arrow or scale bar will raise a dialog asking upon which component the item is to be based. If the layout does not contain any components for which a legend, North arrow or scale bar can be created, these buttons will be disabled.
The default line size of both horizontal and vertical lines is 1/20th of a point. Since this is a thinner line than can be displayed in one pixel on most monitors, keep in mind that the line will print more thinly than it appears on the monitor.

Creating Images from Layouts

Layouts have a special role in creating images because they capture and unify scale information for all of the elements that are used in them. This makes it possible to use layouts to create a single image at a specified resolution even though the layout is composed of drawings or other non-image elements.

To create an image from a layout, open the layout and use the Tools - Make Image command and specify the resolution desired (the default will be the resolution of the screen display, usually 72 or 96 DPI). This will create an image in the project. If an image file is desired, the image may then be exported to any image format supported by Manifold.

Another way of creating an image from a layout is to use File - Export - Image to save the layout as a .emf, .ps or .pdf file. When saving a layout via File - Export - Image we must specify the desired Pixel resolution and the Vector resolution of the file.

Pixel resolution (screen resolution by default) is used to save images and surfaces that are in the layout. Vector resolution (300 DPI by default) is used to save drawings and labels as well as layout elements such as text or horizontal and vertical lines. If saving to PDF for subsequent work in some other program before printing choose a Vector resolution setting that is the same as the resolution of the printer that will be used. If printing to a 600 DPI printer, for example, use 600 DPI. Pixel and Vector resolution must be at least 1 DPI and cannot exceed 7200 DPI.

See the How to Print topic for an example of saving a layout to a .pdf file.

Example

In this example we create a new layout from a drawing.

We begin with a drawing of Mexico. We would like to print this drawing using a layout.
In the project pane toolbar we click on the **Create** button down arrow for a pull-down menu of project pane components that may be created.

![Create button](image)

We choose a **Layout**. This will pop open the **Create Layout** dialog.

![Create Layout dialog](image)

The **Create Layout** dialog displays a list of all components in the project. This project only has one component, the Mexico drawing. We click on **Mexico Drawing** to highlight it and then press **OK**. If we wanted our new layout to be named something other than "Layout" we could have specified a new name as well. We will not use a template. See the Layout Templates topic for information on templates.

![Project pane](image)

In the project pane a new layout is created underneath the parent **Mexico Drawing** component. To open the layout, we double-click on it.
The default layout will show the entire component on whatever is the default page size and orientation defined for our printer. Since most people use Letter or A4 sized paper in a portrait orientation it is quite likely the layout will look like that above. The light gray lines bordering the drawing show the extent of page margins that have been set for this printer.

Given the East - West extent of Mexico, our drawing would look better if printed on the paper in landscape orientation.

To change the paper orientation we choose **File - Page Setup** and click on **Landscape**, then we press **OK**.
The layout will be redisplayed using landscape orientation for the paper. If we like, we can print the layout now with File - Print.

To continue this example we will resize and move the layout element that shows the Mexico drawing component. Before editing the layout we will first change the selection style used. Manifold can show selections in different styles, with the Dense Dots style being the default. It may be more convenient to show selections in layouts using the Border style.

To change the selection style we click on the down arrow of the Selection Style button in the selection toolbar.

We then choose Border.

We can now continue with the example.
Layouts can contain more than one component. When a layout is created from a component as we have done above, it will contain that component as the initial design element in the layout. To select the layout element we **CTRL-ALT** click on it.

The selected element will appear with a red border with editing handles. We can resize the element by clicking on an editing handle and dragging. Dragging on a corner will resize both the horizontal and vertical sides at the same time. We can also resize vertically and horizontally using the vertical and horizontal edit handles.
The resize action will be shown with a moving preview box as we drag the mouse.

Release the drag to finish the resize. The component seen in the layout will also be resized to fit in the maximum vertical or horizontal extent that it can fit.
To move a selected element, click on it and drag.

A preview box will appear as we drag the mouse to a new location.
Releasing the drag will move the element to the new position.

Clicking outside the selected element will deselect it.

**Selection in Layouts**

We normally use smart mouse selection within layouts, although of course the regular selection commands will work as well. Edit menu selection commands such as **Select All**, **Select None** and **Select Inverse** may be used. The **Selections** pane may be used to save selections. Selection styles may be altered. For example, many users will use the **Border** style within layouts.

To edit a layout element, use a **CTRL-ALT** click to select the element for editing. This will make the element the **primary selected** element.
Formatting operations (such as changing the font of a text element or the size of a line) will be applied to the primary selected element if there is more than one element selected. For convenience, if only one element is selected the formatting toolbar will work for that element. This allows rapid clicking on an element to select it followed by formatting.

Rotating a Layout Element

Layout elements are shown “North Up” by default. An element may be rotated by selecting it for editing and then pressing the **Element Rotation Angle** button in the format toolbar.

Consider a layout element that’s been selected for editing.

> We click the **Element Rotation Angle** button to rotate the element.

This calls up a menu of common choices for rotation angle. If we want to specify an angle not in the menu, we choose **More** for a dialog that allows us to specify an exact angle. We choose **40** to specify 40 degrees clockwise rotation.
The element is rotated 40 degrees clockwise. Note that the selection frame and editing handles also rotate. We can drag the editing handles in the new, rotated angular directions.

If we click outside the element to deselect it, we can see it has been rotated 40 degrees clockwise.

**Properties for Drawings and Maps**

How an element appears in a layout is controlled by its properties. Right click on the element and choose **Properties** in the context menu to call up a **Properties** dialog, or simply double click the element. Right click on a text element and choose **Edit** to edit the text.

The **Properties** dialog for a layout element showing a drawing has many properties:
**Properties**

**Scope**  Choose *entire component*, *layer*, *locked rectangle*, *locked center / scale*, *saved selection*, *selection*, or *view* to specify the contents of the element. If views or selections have been saved, choosing *view* or *saved selection* will enable the choice box to the right with all of the views or saved selections available.

**Paging**  Controls the printing of this component within multipage layouts:

- **continuous** - Spread output evenly across the entire range of pages they appear on.
- **individual** - Confining output to only that portion of a page on which the component appears and replicate this output likewise on all pages. Used to create overview maps with a printout repeating on all pages.

**pages or page ranges**  Used with multipage layouts to control which page this element appears in. Enter a series of page numbers or page ranges separated by commas. White space is ignored and reverse ranges are interpreted correctly. For example, we could enter 1, 2, 5-8 or 1, 2, 8-5 and get the same results. See the Multipage Layouts topic.

**Background**  Show the background color for this component, or change it in the layout. Choose *auto*, *custom* or *none*. *custom* allows choice of a custom background color in the adjacent color well.

**Control points**  Display any control points embedded in this component. Choose *auto*, *hide* or *show*.

**Graticule**  Display a graticule overlaying this component. Choose *auto*, *hide* or *show*. A graticule will not be displayed in the layout unless the user has explicitly configured it in the component's window using *View - Graticule*.

**Grid**  Display a grid overlaying this component. Choose *auto*, *hide* or *show*. A grid will not be displayed in the layout unless the user has explicitly configured it in the component's window using *View - Grid*.

**Legend**  Display the legend defined for this component. Choose *auto*, *hide* or *show*. 
North arrow  Display the North arrow defined for this component. Choose auto, hide or show.

Scale bar  Display the scale bar defined for this component. Choose auto, hide or show.

Border  Show a border for this component. By default, no border is shown. Borders are drawn using the color shown in the adjacent color well, black by default. Choosing coordinates, coordinates (graticule) or coordinates (grid) for a border style will print labels along the edges of the border. The default font is set by the Border entry in the Tools - Options - Fonts dialog.

[color well]  Choose a color for the border.

ABC  Enabled when using a coordinates border style. Double-click to choose the font to be used for labels.

Size  Enabled when using a coordinates border style. Specify the width in points of the border margin in which coordinate labels are shown.

Interval  Enabled when using a coordinates border style. The stepping between label values and the unit of measure to be used.

Rounding  Enabled when using a coordinates border style. A formatting option that rounds values for the coordinate labels as specified. For example, rounding to 1 digit will round down to the nearest ten with values like 115 rounded to 110 and 95 rounded to 90. The main usage of this formatting option is to suppress unwanted trailing zeros in minutes and seconds when using degrees - minutes - seconds format. Setting the Interval to a fractional value such as 1.5 will automatically adjust Rounding. Choosing arc minute or arc second as the unit in Deg-min-sec format will also change Rounding automatically so that the specified units will appear with full precision.

Deg-min-sec format  Enabled when using a coordinates border style. Toggle between decimal degrees and degrees, minutes and seconds format when degree coordinates are used. Currently enabled only when the layout component is in Latitude / Longitude form. The default is to use decimal degrees format.

Resolve Overlaps  Enabled when using a coordinates border style. Automatically clip labels when so many are specified that they would overlap.

Border each page  Enabled when using a border style, for use when working with continuous style paging in multipage layouts. When checked, draws a border for each individual page as would be used for map books. When not checked, draws a single continuous border as would be used when the multipage printout is intended to be tiled together into a single large print.

Accessory visual elements such as legends, North arrows or scale bars are defined within the component shown in the layout. The Properties dialog allows us to specify how (and if) these will appear in the layout.

auto  Show if the item is turned on in the component window, otherwise hide.

hide  Always hide the item.

show  Always show the item.
For example, we might not have a scale bar turned on in a drawing when we open it in a drawing window but we might want to have the scale bar turned on in a layout. In that case, we would choose show to show the scale bar in the layout whether or not it is turned on in the drawing window.

**Labels on Borders**

A common task is to place labels along the borders of a layout element that show the latitudes and longitudes of locations within the element.

This is a two-step process in Manifold: first a graticule is configured for the element within its own window and then the graticule is used to guide the creation of labels in the layout window. Graticules are used to guide the placement and values of border labels because graticules provide a simple, visual way of choosing reasonably even intervals at which to show latitudes and longitudes.

**To create latitude / longitude labels on the border of a layout element:**

1. Suppose layout element in question is a drawing. Open the drawing in its own window.
2. Use View - Graticule to turn on a graticule and to configure it as desired.
3. In the layout window, right click on the drawing and choose **Properties**.
4. In the **Properties** dialog, choose **coordinates (graticule)** for the **Border** style.
5. Check the **deg-min-sec format** box if degrees, minutes and seconds format is desired.
6. Press **OK**.

Labels will appear along the border of the element at intervals and values aligned to the graticule lines. The labels will appear when the **coordinates (graticule)** option is selected in the **Border** style box. The graticule lines themselves will appear or not appear depending on the setting of the **Graticule** option. It is possible, therefore, to show latitude / longitude labels on the border based on the graticule while choosing **hide** for the **Graticule** option and thus not showing any graticule lines.

Graticule and grid labels will not be displayed in a layout unless a graticule or grid has been explicitly configured within the component’s window using the **View - Graticule** or **View - Grid** commands.

When using **coordinates (graticule)** the graticule labels will be placed along all four edges of the layout entry as needed whether they are latitudes or longitudes. Note that latitude labels might appear on either upper and lower edges or, if need be, on the left or right edge of the layout.
With some projections as shown in the illustration above meridian lines will cross an upper or lower border edge as well as a left or right edge. It is possible in such cases to have one label for a meridian at the top edge of a layout and another label for the same meridian at the right edge of the layout. Manifold will automatically place graticule labels correctly in such cases.

When overlap resolution is turned on, vertical edges give priority to latitudes and horizontal edges give priority to longitudes. The locations of graticule labels in layouts for projected components are accurate to approximately 1/300th of a page.

See the Graticule Labels in a Print Layout example topic for creation of graticule labels.

**Border options**

In the Properties dialog for a layout element such as a map, drawing, image, labels component or surface the following options are available for Border styles:

- **coordinates** Print border labels using the native coordinates of the component. Projected drawings, for example, using projections with meters as units of measure will have labels in meters. Lat/lon drawings will have labels in degrees.

- **coordinates (graticule)** Print border labels with position and values aligned to graticule lines. Requires configuring a graticule first for the component in the component's window using View - Graticule. The preferred way of creating latitude and longitude border labels.

- **coordinates (grid)** Print border labels with position and values aligned to grid lines. Requires configuring a grid first for the component in the component's window using View - Grid. The preferred way of creating border labels at even intervals using native coordinates in projected drawings.

- **none** Do not show a border.

- **thick** Show a rectangular border using a thick line.

- **thin** Show a rectangular border using a thin line.

See the Graticule Labels in a Print Layout example topic for use of the above options.

**Properties for Text Elements**
Some layout components will have a simpler **Properties** dialog. For example, a text element will have just a few options in its **Properties** dialog:

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bound to</td>
<td>The component from which this text element takes its system-generated text expressions.</td>
</tr>
<tr>
<td>Page filter</td>
<td>Used with multipage layouts to control which page this element appears in. Enter a series of page numbers or page ranges separated by commas. White space is ignored and reverse ranges are interpreted correctly. For example, We could enter 1, 2, 5-8 or 1, 2, 8-5 and get the same results. See the Multipage Layouts topic.</td>
</tr>
<tr>
<td>Border</td>
<td>Show a border for this component. By default, no border is shown. Borders are drawn using the color shown in the adjacent color well, black by default.</td>
</tr>
</tbody>
</table>

When a new text element is inserted into a layout that contains only one component element, the text element will automatically be bound to that component. If there are several components already in the layout the text element will be bound to none.

**Views in Layouts**

When the scope of a layout element is set to **View**, the layout element will use the view to guide what is displayed within the layout element. A view specifies a central point and a scale, but views themselves are limitless in extent. How much of the view is seen depends on the size of the layout element. If the element is resized smaller, less of the view will be seen. If the element is resized larger, more of the view will be seen.

**Tables in Layouts**

Layouts may be created from tables, and tables may also be dragged and dropped into a layout. See the Printing Tables topic for a discussion of tables and layouts.

**Layouts with multiple elements**

More than one component can be dragged and dropped into a layout.
Suppose we've created a layout using a drawing of Europe. We CTRL-ALT click on the Europe element in the layout to select it for editing.

We can now resize it by dragging the edit handles in the corners and the sides. For example, we could resize it by dragging on the lower right corner's edit handle to make the element smaller.
For example, we could resize it so that it fits into the upper left corner of the page. Click anywhere else in the layout to deselect the drawing.

To add another component we drag and drop the component from the project pane into the layout. For example, suppose we have the sample Bronze image in our project. We can click on it and drag it from the project pane into the layout.

When dragging into the layout the Bronze image will be dropped where we release the drag.
The image appears in the layout where we drop it. It appears using a default size. We **CTRL-ALT** click on the image to select it for editing.

We can now resize the image by dragging the upper left corner edit handle.
We can then click and drag the image to move it to the upper right corner of the layout.

Click on any empty space to deselect the image.
Suppose our project also contains the Schloss sample image. We can add it to the layout by dragging and dropping it from the project pane into the layout.

As before, the element is dropped into the layout at the point we release the mouse button on the drag. The new image appears in a default size. We will CTRL-ALT click on the Schloss image to select it for editing.
We can now resize the image by dragging any of the edit handles.

After resizing, we can move this image by dragging it to a new location.
In the illustration above, we've moved the Schloss image so that it partially overlaps the Bronze image in the layout.

If desired, we can move the Schloss image below the Bronze image so that the Bronze image is not partially covered up. To do so, we right click on the Schloss image and from the context menu choose Order - Move to Bottom.

This will move the Schloss image to the bottom of the layout element stack so that it no longer covers up part of the Bronze image.

**Aligning Items in Layouts**

Items may be aligned in layouts by selecting them and then using the alignment toolbar to invoke commands that align those items. The alignment toolbar is turned off by default. To turn it on, open a layout, choose Tools - Customize and check the box for the alignment toolbar. Alignment toolbar commands allow us to line up items in the layout with each other, to resize them, to center them and otherwise place and size them as desired.
Suppose we have a print layout with two elements, a drawing and an image. We clicked on the drawing to select it and then SHIFT-clicked the image to also select it as the primary selected item.

Clicking **Align Bottom** in the alignment toolbar will align the drawing to the bottom of the image. The alignment controls work by aligning all other selected items to the **primary** selected item (the one with the edit handles). See **Smart Mouse Selection** for quick moves to designate an item the primary selected item. See the Align Items in Layouts example topic.

Commands from the alignment toolbar are repeated in the **Layout** menu **Align, Center, Space, Resize Full** and **Resize Same** commands. The **Layout** menu appears when a layout window has the focus.

**Repositioning Multiple Items in Layouts**

Clicking and dragging the primary selected item in a layout will move that item. **SHIFT** clicking and dragging the primary selected item in a layout will move that item and will also move all other selected items as well.
Creating Frames and Borders with Lines

Use the horizontal line and vertical line tools to create borders and frames to visually organize other elements in the layout.

**Insert Horizontal Line**
Create a horizontal line in the layout. The default line size is 1/20th of a point.

**Insert Vertical Line**
Create a vertical line in the layout. The default line size is 1/20th of a point.

**Insert Box**
Create a rectangular box with a click and a drag. Boxes have foreground and background color and line thickness that can be set by selecting the box and using the format toolbar. The default line thickness is 1/20th of a point.

**Insert Box on Center**
Create a rectangular box centered on initial mouse click.

To assure lines start and stop at exactly desired locations, use a grid and **Snap to Grid**.

Choose View - Grid to launch the Grid dialog. Check the Show grid box to turn on the grid. Grids are created on a 1-centimeter spacing for layouts by default. Choose the X (horizontal) and Y (vertical) spacing and units desired and press OK.

The result is that a grid of dots appears within the layout.
Turn on Snap to Grid by first showing a grid with the View - Grid command and then pushing the Snap to Grid button or by choosing Edit - Snap to - Grid from the main menu. When Snap to Grid is on, the mouse cursor will move only between grid points on the form. Lines may therefore be created with end points that are snapped exactly to locations marked by grid points. See the Edit - Snap To topic for additional details and extra commands.

We can then use Insert Vertical Line and Insert Horizontal Line to draw lines for borders and other boxes.

Seen without the grid turned on, this is a typical artistic arrangement for printing, where the subject of the map is in the upper area and the three bordered boxes along the lower edge might be filled, for example, with a logo, the title of the map and some additional information.
Seen zoomed in, we’ve added a map component and have added a text element with two lines in it.

The **Snap To Elements** button causes the cursor to snap to the corners of existing elements in the layout. This is a convenient way of aligning or resizing elements to match exactly the dimensions or edges of other elements.

**Transparent Colors**

Layout elements use transparent color by default for a background color, thus giving a “floating” effect so that whatever items are in the background can be seen through the element’s “white space.” Transparent background color is often used with text labels in layouts. Exceptions are vertical and horizontal lines, which do not have a background color and which do not allow transparent foreground color.

**The Layers Pane and Layouts**

When a print layout window has the focus the Layers pane will show each item in the layout as a layer. Available commands include:

- Checking a box will turn the layout element on and unchecking the box will turn the layout element off. Hiding an element by unchecking its box will prevent it from being clicked, selected or printed.
- Moving “layers” up and down in the layers pane will move the layout elements up and down relative to each other within the layout.
- Double clicking onto a text label in the layers pane will allow immediate editing of the text in the box directly from the layers pane.
- The **Delete Layer** button in the layers pane toolbar will be enabled when any layer except the main layer is clicked. Deleting a layer will remove this element from the layout.
- The **Properties** button in the layers pane toolbar will open the properties dialog for any component-based element in the layout.

The layers pane is a key control when working with complex layouts.
In the example above we see a layout together with the layers pane. The layout is based on a main element that is a drawing of Europe. Below it is a text element that consist of the phrase "Copyright 2002" repeated several times in an outline font using a light gray color. Above the main element is another text element giving the title, "Europe," and two images, the example Bronze and Schloss images. The Background layer is turned off, so behind the layout page we see the checkerboard pattern used in Manifold when there is no background.

Note that the layout elements added with layout tools, such as text, are shown in the layers pane with special "layout element" icons. Text elements will be named using the initial characters of their text. Components will be shown with their scope, so that if a component is used more than once with different scope (for example, with different views) it is easy to tell which element is which in the layers pane.

With many overlapping elements in a layout we can use the layers pane to turn off upper elements if we want to select or otherwise work with lower elements.

**Layouts in the Layers Pane with other Components**

When a drawing or map or other component from which a layout has been created has the focus the layers pane will show layouts as "layers" in the layers pane. Checking the box for one of these print layout layers will cause a layout rectangle to appear in the component that shows the region covered by the layout.
In the illustration above the drawing has four layouts that show different parts of Mexico. Three of the layouts have been checked in the layers pane causing three layout preview rectangles to appear in the drawing.

The preview layout rectangles will be automatically adjusted if we change the scope of the layout.

For example, suppose we open a drawing window and a layout window for the same drawing and show the layout in the drawing window by checking the layout layer's checkbox. We select an area in the drawing by clicking on it with Select Touch. The scope of the layout (set by right clicking the layout in the layout window and choosing Properties) is set to entire component.

If we now change the scope of the layout to be the selection, the layout will show only the selected objects (one area, Germany, in this illustration) and simultaneously the drawing window will be updated to show the new region covered by this area.

Right clicking onto the hatched border of one of the layout rectangles in the drawing will cause a context menu to appear with controls based on that layout rectangle. For example, we can Zoom to a given layout rectangle, Print it or change its Properties. If a layout is empty (for example, if the layout scope is set to selection and nothing is selected in the parent component) zooming to the layout will do nothing.

Use Tools - Options - Colors - Layout Rectangle to change the color in which layout rectangles are shown. The default color is black.
We can insert text into the layout using the **Insert Text** button. Click on the **Insert Text** button and then draw a text box in which the text will be located. In the **Insert Text** dialog, enter the text that is desired. See the How to Print topic for a detailed example of adding text to a layout.

After creating a text element in a layout we can later edit the text by double clicking on the text element.

**System Generated Expressions for Text**

The **Insert Text** dialog includes an upper pane that lists a hierarchy of available expressions that may be used to automatically insert system-generated text into a text element. Double-click onto an expression to add it to the text element. Values for the text element, like scale, are taken from the main element in the layout.

The **Trim numbers** option will automatically trim numbers generated by the system to the specified number of decimal digits. This is very useful when creating latitudes, longitude or other numbers and many digits to the right of the decimal point are not desired.

Let’s consider an example. We will work with a project that has been saved in a file called *Eurasia.map*. We would like to add the name of the file to a layout.

To do so we add a text element to the layout. We then click into the **Insert Text** dialog’s lower pane and enter “Hello, my name is ”. In the upper pane we can click on the **Project** folder to expand it for a list of available expressions related to the project.
If we double-click onto the [Filename] expression it will be added to the text being assembled in the lower box. We can then click into the lower pane and add a period "." to finish the sentence.

If we double-click onto the [Filename] expression it will be added to the text being assembled in the lower box. We can then click into the lower pane and add a period "." to finish the sentence.

Hello, my name is [Filename].

After we click OK we can see how the text appears in the layout. Note that [Filename] has been replaced with Eurasia.map, the name of the file.

The following expressions may be inserted into text elements. They will be replaced automatically with the corresponding text whenever the layout is refreshed. The values are taken from the parent component of the text element. To change the parent component of a text element, right click on the text element, choose Properties and then choose the component desired in the Bound to box.

Note: The names of the system-generated expressions are case sensitive. For example, [date] does will not work as a substitute for [Date].

Component

[Average Height] Available for text elements bound to Profiles and Elevations elements in the layout. Gives the average height over the surface for the profile line. Profiles and elevations are optional components that are available when the Surface Tools package is installed. See the Profiles and Elevations topic for more information.

[Component] The name of the parent component in the project.

[Coordinate System] The coordinate system in use for the parent component (same as [Projection]).

[Datum] Name of the datum in use in this projection / coordinate system.

[Description] The contents of the parent component's Description property.

[Extent English] Scale reported as horizontal extent in English units.
[Extent Metric] Scale reported as horizontal extent in metric units.

[Maximum Height] Available for text elements bound to Profiles and Elevations elements in the layout. Gives the maximum height over the surface for the profile line.

[Minimum Height] Available for text elements bound to Profiles and Elevations elements in the layout. Gives the minimum height over the surface for the profile line.

[Page Center] Center of the page in projected coordinates.

[Page Center LatLon] Center of the page in latitude, longitude coordinates.

[Page Center LatLon DMS] Center of the page in latitude, longitude coordinates using degrees, minutes and seconds notation.

[Page Center Latitude] Latitude coordinate of the center of page in latitude, longitude coordinates.

[Page Center Latitude DMS] Latitude coordinate of the center of page in latitude, longitude coordinates using degrees, minutes and seconds notation.

[Page Center Longitude] Longitude coordinate of the center of page in latitude, longitude coordinates.

[Page Center Longitude DMS] Longitude coordinate of the center of page in latitude, longitude coordinates using degrees, minutes and seconds notation.

[Page Center X] X coordinate of the center of page in native units.

[Page Center Y] Y coordinate of the center of page in native units of the component being printed.

[Projection] The projection in use for the parent component.

[Scale] Printing scale in absolute scale style, for example 1:10000.

[Scale English] Scale in English units using relative scale style, as in 1 ft : 100.00 miles.

[Scale Metric] Scale in Metric units using relative scale as in 1 cm : 100.00 km.

[Scale Numeric] Scale as a 1:N ratio, same as [Scale].

[Unit] Native unit used in coordinate system / projection.

**Date and Time**

[Date] Current date in short date format specified in Windows Regional Options setting. A typical Windows default is “8/27/2003”.

[Day] Current day in the month from 1 to 31.

[Hour] Current hour from 1 to 24.

[Long Date] Current date in long date format specified in Windows Regional Options setting. A
Printing and Layouts

typical Windows default is "Wednesday, August 27, 2003" format.

[Minute] Current minute, from 1 to 60.

[Month] Current month, from 1 to 12.

[Month Abb] Month as a three-letter abbreviation, such as "Aug".

[Month Name] Month name, such as "August".

[Second] Current second, from 1 to 60.

[Time] Current time in format specified in Windows Regional Options setting. A typical Windows default is "3:41:08 PM".

[Weekday Abb] Day as a three-letter abbreviation, such as "Wed".

[Weekday Name] Day name, such as "Wednesday".

[Year] Current year.

Layout

[Layout] Report the name of the layout.

[Layout Description] Report the description of the layout.

Print Job

[Computer] Windows name of the computer executing the print job.

[Page] Page number as printed.

[Page Above] In multipage layouts, the page number of the page above this page. If no page exists above this page, prints "none".

[Page Below] In multipage layouts, the page number of the page below this page. If no page exists below this page, prints "none".

[Page Left] In multipage layouts, the page number of the page to the left of this page. If no page exists to the left of this page, prints "none".

[Page Right] In multipage layouts, the page number of the page to the right of this page. If no page exists to the right of this page, prints "none".

[Pages] Total number of pages in the print job.

[Page X] When printing multipage layouts that are to be tiled together, the position from left to right of this page.

[Page Y] When printing multipage layouts that are to be tiled together, the position from top to bottom of this page.

[Printer] Windows name of the printer.

[Printer Port] Windows name of the printer port.

[User] User login running Manifold.
Project

[Filename] Name + extension of the project file. In a new project, will be empty until the project is first saved.

[File Path] Fully qualified name of the project file. In a new project, will be empty until the project is first saved.

[File Title] Title of the project (displayed in the main window title). Before a project is saved for the first time will be something like "Project1". After the project is saved, will be the same as the project file name without the .map extension.

Square brackets [ ] may be used in running text within text elements as long as the square brackets are not used exactly in one of the above escape sequences (in which case the escape sequence will be replaced by the information it contains. Therefore, one can create a text element in a Manifold layout such as

Figures shown as [123] values are elevations.

If we don't like the default [Date], [Long Date] and [Time] formats used by Windows we may easily change them using the Control Panel's Regional Options applet. For example, we can change from AM/PM designations to 24-hour time, remove the day of the week from the [Long Date] and so on.

Scale text is reported in either relative scale or horizontal extent styles similar to those available for use in the status bar scale report. Relative scale reports the ratio between one centimeter or inch and the scaled number of meters/kilometers or feet/miles as in, for example, 1 cm : 100.00 km. Horizontal extent shows the horizontal size of the main layout entry.

Note that in the hierarchical list of system-generated expressions, the component folder will only be available for text entries belonging to layouts that were created from a component in the first place; otherwise, it would be unclear what component is being referred to.

Multipage Printing

A layout can be expanded to use more than one page with Layout - Pages and specifying the number of pages in the horizontal (x) and vertical (y) directions. Elements in the layout cannot be resized across page boundaries, although a single element can cover the entire multipage layout.

Using Maps or Using Layouts

For sophisticated compositions there may be a choice as to whether to create the visual effect desired in the layout by first composing multiple layers in a map and then using that map in a layout, or composing multiple elements in a layout. We might use a combination of both approaches.

For example, one can create a title block either in a layout or in a map. Suppose we want to create a rectangular box that contains a main title. We could create this in a layout by inserting a box using the controls in the Tools toolbar and then using the Insert Label command from the toolbar to add a text element.

We could also achieve the same result by creating a map that had three layers consisting of two drawings and a labels component. For example, suppose we wished to print a map that included a title caption with the text "Europe" in a box.
We could create a map that provided this caption.

The map could consist of three layers as seen above. The bottom layer is simply a rectangular area colored white. If we drag and drop this map into a layout, the white rectangle will provide a blank white background for the title text no matter what is underneath in the layout.

This same caption could have been created using the print layout *Insert* tools described at the beginning of this topic. For simple captions we would no doubt create the title block using the layout. For more elaborate effects, such as the use of Gaussian drop shadows, we would use the map.

### Printing to Scale

Printing to scale is accomplished by first creating a view at the desired scale and then printing that view within a layout.

**To print at a given scale:**

1. Open the map you want to print. Make sure it is a projected map.
2. Use *View - Zoom To* for setting the desired scale, for example *1:3000000*.
3. In the Views pane, create a new view. Rename the view to a memorable name such as "New York 1:3000000".
4. Create a layout component. Open the layout window and drag and drop the map component from the project pane into the layout window.
5. Right-click the map image within the layout window and select *Properties*. Set *Scope* to *View* and select the view created in step 3. Press *OK*.

When printing to scale, make sure the view selected is centered upon the area of interest. The actual extent of the component visible in the layout element will be controlled by the scale and by the size of that layout element on the page.

### Status Bar and Scale in Layouts

The status bar for layouts will report mouse cursor position in either inches or millimeters depending on the value of the *Use English measurement units* option in *Tools - Options - User Interface*. Page numbers will be added if a multipage layout is used.

The scale readout in the status bar gives the scale of the layout (in actual paper size) relative to the screen display. If the scale readout is set to use absolute scale (as in "1:100") and the layout is zoomed using *View - Zoom to* into a zoom of 1:1, then the layout will appear on the monitor in exactly the size that it will be printed. The status bar readout will also be 1:1.
For example, suppose we have a layout that uses one sheet of Letter size (8.5 x 11 inches) paper in Landscape mode and we have a very large, high-resolution monitor. We can zoom in to 1:1 scale in the layout and it will appear as an 8.5 inch by 11-inch sheet on the monitor.

Note that the scale reported in the status bar is not the geographic scale of what is in the layout but rather the scale of the layout as it would be printed on real paper compared to what is shown in the monitor. To get the geographic scale of the component as it will be when printed, insert a text element and use one of the system generated text expressions, such as [Scale] to report a scale in the text.

**Using Layouts Templates for Repetitive Work**

If we create many print jobs we may want to use the same type of layout for more than one job, repeating standard border arrangements. The best way to accomplish this goal is to use layout templates. Layout templates are by far the fastest and easiest way of recycling the look and feel of layouts into other layouts. If you create a lot of similar layouts, it is wise to master the use of layout templates. See the Layout Templates topic for details.

**Copy and Paste**

If desired, we can use Copy and Paste to move layouts between different instances of Manifold. Launch Manifold and open the project in which a layout is saved. Launch a second instance of Manifold in which the working project will be done. Copy the layout from the first instance of Manifold and paste it into the second instance of Manifold.

When copying and pasting layouts between different instances of Manifold all layout elements for components along with all dependent legends, North arrows, scale bars and text elements bound to a specific component will be removed. Nonetheless, the copy and paste method is very useful for many types of simple layouts consisting of boxes and vertical and horizontal lines and generic text such as copyright notices, statements of legal policy and so forth.

Using Copy and Paste to make copies of layouts within the same project is a good way to avoid irreversible errors when altering complex layouts. When a layout gets to the point that an error would be inconvenient to reverse, make a copy of the layout and continue working with the copy.

**Layouts and Enterprise Edition**

When using Enterprise Edition layouts can be shared to an Enterprise server. The Share command is enabled even if the main component in the layout is not shared. However, when sharing a layout, the system will automatically share all components used by that layout to the same Enterprise server if they are not yet shared. Note that it is possible to have an unshared layout that includes a shared component but it is not possible to have a shared layout that includes an unshared component.

**Notes**

Layouts may be copied and pasted, and layouts may be imported from a different project by using File - Import - Component.

The example date used for the system generated date, Wednesday, August 27, 2003, marked the closest approach between Earth and Mars in approximately 73,000 years.

When used with layouts, the tracker tool will report in either English (inches) or metric units depending on the setting of the Use English measurement units option in Tools - Options.

**Tech Tip: Printing a Layout to PDF**

There are two ways to create a PDF file from a layout: exporting a layout to a PDF or "printing" the layout using a PDF virtual printer driver, which mimics a printer to capture any printing activity as a PDF.

Printing a layout to a virtual printer driver that produces PDF feeds the layout through an ordinary GDI / GDI+ rendering pipeline, which issues "dumb" printing commands that are then captured by the driver. Exporting a
layout directly as a PDF feeds the layout through the Manifold PDF rendering engine, which optimizes printing commands. In general, the results of exporting a layout as a PDF will almost always be better than the results of printing the same layout to a virtual printer driver.

See Also

Exporting Layouts
View - Zoom
View - Graticule
View - Grid
View - Legend
View - North Arrow
View - Scale Bar
Graticule Labels in a Print Layout
Align Items in Layouts
Layout Templates

Layout templates are pre-defined arrangements of layouts that may be invoked when creating a new layout or applied to an existing layout.

- To use a template when creating a layout, in the Create Layout dialog check the Template box and choose a template.

- To apply a template after creating a layout, open the layout and use the Edit - Template - Apply command to use a built-in template or a template added to the system through customization, or use the Edit - Template - Apply File command to apply a template from a file.

- To create a new template, create a layout and then use the Edit - Template - Save command to save the layout to a file for use as a template.

- The Edit - Template - Apply command also allows us to use a layout in our project as a template to be applied to the current layout.

Standard Templates

There are two built-in templates:

- **Simple, Entire Component** - Shows the entire parent component with the component name as the main title and several information lines.

- **Simple, View + Overview** - Similar to the above, but shows a view named Layout in the parent component as the main element with an inset element showing the entire component to give an overview. If there is no view named Layout that has been saved in the Views pane for the parent component, then the main element will be blank.

Templates try to provide as much information as possible to duplicate the look and feel of a given layout arrangement. They remember the basic arrangement of elements including their colors and other format options such as borders used in text elements. They save element visibility and preserve Z order, that is, the order in which elements are stacked above each other.

Example

Consider a map of Mexico that’s been thematically formatted. To create a print layout, we right click on the drawing in the project pane and choose Create - Layout.
In the Create Layout dialog we check the Template option and choose Simple, Entire Component as the template we would like to use.
When we open the resulting layout we see that the result is a neatly-arranged layout that includes a view of the entire component with a North arrow, a main title and a block of additional text information.

The block of supplementary text information provides the projection in use, the central latitude and longitude and the current date.

Templates consist of pre-arranged elements for a layout. We can edit them just like any other element.
For example, if we are curious as to how the additional text information is created we can double-click on it to see its properties. As we can see from the above, it is simply a text element that has a variety of escape sequences in it to have text information automatically loaded from the parent component.

The **Simple, View + Overview** layout template works with drawings that have a previously-saved view called **Layout**. To see this template in action we begin by zooming into the Mexico drawing near the province of Durango.
In the Views pane we create a view called Layout. We right click on the drawing in the project pane and select Create - Layout. In the Create - Layout dialog we check the Template box and choose the Simple, View + Overview layout template.

In the project pane we right click on the Mexico drawing and select Create - Layout. In the Create - Layout dialog we check the Template box and choose the Simple, View + Overview layout template.
When we open the resulting layout we see it has a main element that shows the Layout view zoomed in to the province of Durango as the main element as well as an inset smaller element that shows the entire component to provide a contextual overview.

**Using Edit - Template - Apply**

The **Edit - Template - Apply** command allows us to apply either existing templates or existing layouts to a layout. Open the layout to be operated upon, choose the command and then choose either an existing template or an existing layout to apply to the layout.

**Example**

Let's create a generic layout and then use it as a template for a different layout.
We begin with the Mexico drawing illustrated in the examples above. We create a layout called **Layout 3** from the Mexico drawing. In the **Create Layout** dialog we don't use any other templates. It doesn't matter whether we do or do not use a template, but for the sake of this example we will create a very simple layout to show that what we create, even if we don't use another template, can be used in turn with another layout.
After some tinkering and editing we create the above layout. It shows the parent component within a box border. Large captions are above and below the main element with a small scale bar to the right.
So far, so good. We now change gears and import a drawing that shows congressional districts in the US. We create a simple layout for the drawing as seen above.

To apply an existing template or layout to this layout we choose **Edit - Template - Apply**. In the **Apply Template** dialog we check the **Apply a template** radio button and choose the **Simple, Entire Component** template. Press **OK**.
The result is that the layout showing congressional districts immediately switches to the arrangement specified by the Simple, Entire Component template. Excellent!

To change the arrangement again, we once more choose Edit - Template - Apply.

In the Apply Template dialog we check the Apply a layout radio button and choose the Layout 3 layout we created earlier. Press OK.
The result now is that the layout arrangement used in the Layout 3 component has now been applied to this layout. It is, of course, somewhat absurd to provide a title of "Mexico" to a layout that shows US congressional districts, but this helps to illustrate the idea of using one layout to specify the arrangement of a different layout. The Edit - Template - Apply command makes it very easy for us to recycle the work done to create clever templates.

Obviously, if we wanted to create a series of templates to provide a standardized look and feel for the print layouts we create we would take care to use escape sequences for things like component names, so that the resulting layouts are automatically customized with correct names for the components they show.

Note, by the way, that if we are going to use a scale bar in a template to be used with many different components it makes sense to think carefully how we specify the scale bar. In the above illustration a scale bar that was defined for use with Mexico doesn't work very well with the US.

**Conveniences**

Manifold will remember the last used layout template and will automatically select it as the default choice in the Create Layout dialog and in the Apply Template dialog. Manifold remembers the last used layout template between Manifold sessions as well.

**System Customization**
If we create a custom layout template `.xml` file and save it in our Manifold System `Config` folder the layout will become automatically available for use as a built-in template. On startup, Manifold will automatically load any custom layout templates found in `.xml` files in the `Config` folder.

**Programming**

The `Document` object includes means to create a new layout using a layout template. The `Layout` object includes means to apply a layout template to the layout, to generate a layout template from the layout and to save the layout as a layout template.

**See Also**

- Layouts
- Custom Layout Templates
- Edit - Template
Specifying Views in Layouts

Manifold provides many ways of specifying the content of a given element in a layout. Once a layout has been created using a component such as a drawing, or if a new element has been dropped into an existing layout, the contents shown by that component are specified by the **Scope** setting in that element's **Properties** dialog.

Think of elements in a layout as viewports or windows that take their content from the component used to create that element. The scope setting tells Manifold what part of the component is to be displayed within that viewport.

The **Scope** setting allows choice of:

- **entire component** - Show the entire component within the element.
- **layer** - If a map is shown, show an entire layer within the map. Equivalent to using **entire component**, but guided by the extent of the specified layer.
- **locked rectangle** - Show a specified region, re-scaling the view as necessary so the region fits within the element.
- **locked center / scale** - Show as much of a region as can fit within the element given the scale used. This option is similar to a **view**, but can be specified without creating a view.
- **saved selection** - Show a saved selection, re-scaling the view as necessary so all items in the saved selection fit within the element.
- **selection** - Show the current selection, re-scaling the view as necessary so all items in the current selection fit within the element.
- **view** - Using the scale and center point of a view from the **Views** pane, show as much of the view as can fit within the element given the scale used.

The scope setting provides several options because sometimes we would like to display the contents of an element at a specified scale, as in printing to a desired scale, and sometimes we don't care what scale is used so long as a specific region of interest is displayed.

**Examples**

In the illustrations that follow we show a drawing window and a layout window next to each other. The drawing shows provinces in Mexico. The drawing has been projected into **Orthographic** projection so that scales and other values may be computed accurately.

The layout was created from the drawing using default settings as described in the Layouts topic. The **File - Page Setup** command was used to switch the orientation of the layout page from **Portrait** to **Landscape**.

**Entire Component**

When a layout is created from a component one element is created that fills the page. The default setting for the scope is **entire component** so that the full extent of the component is shown on the available page. If we resize
the element to be smaller the contents it shows will automatically be re-scaled so that the entire component is always in view.

For example, to resize the element we can **CTRL-ALT** click on it to select it for editing. We can then drag the editing handles, like the lower left handle, to resize the element.

No matter how we change the size of the element window, it will always show the entire *Mexico Drawing* component.

Here we have deselected the element after resizing it. We then right clicked on it, chose **Properties** and then chose a **thin** border for the element. The border makes it easy to see the size of the element without having to select it.
Views

A view in Manifold is defined in the views pane and is used to show a particular region at a particular scale. Views are handy ways of recalling a particular view of a region regardless of the size of the window that is being used.

We can create a view by panning and zooming the drawing window to show a view of the province of Durango.

After panning to the province of Durango and zooming in to approximately the desired view, we then opened the View - Zoom to dialog and zoomed to exactly 1:10,000,000 scale (one to ten million scale). At this scale, one centimeter on the monitor is equivalent to ten million centimeters, or 100 kilometers, in the real world.

Within the views pane we can create a new view, called Durango 1:10m scale to remind ourselves what the view is and what scale it uses.
To change the layout so that it uses this view instead of the entire component, we right click on the layout element and choose Properties. Within the Properties dialog we choose view and then choose Durango 1:10m scale as the view to use. Press OK.

Alternative: An alternate approach is to use locked center / scale instead of view as the scope.

To do so, within the Properties dialog we choose locked center / scale and press the Edit button.
Within the **Edit Center / Scale** dialog we choose the **Durango 1:10m scale** view and press **Apply** to apply the values of this view. **Note:** the **Scale** value shown at the bottom is the internal scale metric used by Manifold to convert from geographic units to dimensionless internal units. The internal metric works all of the time, even if a component has not yet been projected. It is not the same "scale" as meant by the geographic scale of 1 to 10 million.

Why would we use the roundabout method of choosing **locked center / scale** instead of **view**? In some cases, we might want to achieve the same effect as a view even if a view has not yet been defined for the drawing. In the current case, since we are using the **Durango 1:10m scale** view as a fast way of setting the desired center and scale values it does not make sense to use **locked center / scale**. It is faster in this case to simply use **view** as the scope.

The result in either case is that the element viewport in the layout shows the **Durango 1:10m scale** view. Note that more of the view can be seen within the element viewport than appears in the drawing window, because the element viewport on the sheet of paper is much larger than the drawing window on the monitor.

If we were to print out the sheet of paper in the layout and hold it up to the monitor, we would see that the size of the province of Durango is, in fact, the same size as it appears within the rather small drawing window on the monitor.

One way to see this is to zoom the layout window to **1:1 scale** using **View - Zoom To**.
If the layout window is zoomed to 1:1 scale what appears in the layout window will be the same size as it will be when printed out on a sheet of paper. We can see that the province of Durango is, in fact, the same size in the layout as it is in the drawing window.

Note: If we wanted to show in the layout only that part of the view that is visible in the drawing window we could use the locked rectangle scope option. See the discussion of the locked rectangle option below.

Saved Selection and Selection

These scope settings are used to show a component where the visible part is automatically re-scaled so that a given saved selection or the selection is visible.

Suppose we go Back one scene in the layout window (using the left arrow in the navigation toolbar) so that it is no longer zoomed 1:1 but instead the entire sheet of paper is visible, and suppose we go back one scene in the drawing window so that all of Mexico is visible as well.

Suppose we select Mexican provinces in the drawing that border the Sea of Cortez. Note that making a selection in the drawing will also appear in red selection color in the layout as well.
In the selections pane we can save these as a selection called Sea of Cortez.

To tell the element to show the saved selection, we right click on the element in the layout, choose Properties and in the Properties dialog choose saved selection as the Scope and choose Sea of Cortez as the saved selection to use.

The result is that the layout element automatically pans and zooms (re-scales) so that the given saved selection is visible. Note that also will be visible those other parts of the Mexico drawing that will fit within the element's view given the size and shape of the element.
Note also that the saved selection does not have to be chosen as the selection for the saved selection scope option to work. In the illustration above, we have deselected all items (so that there is no current selection), but we can still choose Sea of Cortez as a saved selection for the scope and achieve the same view in the layout.

**Locked rectangle**

Recall that when we used a view of Durango province that the layout element did show Durango at the specified scale, but that because the element viewport was much larger than the drawing window used to create the view much more of Mexico in addition to Durango was visible. Suppose our objective is to show just the province of Durango in the layout, and we really don’t care whether or not the scale remains fixed at one to ten million. We can accomplish this by using the locked rectangle option for scope.

To begin with, let’s reset the scope to entire component so that all of Mexico is visible in the layout element.

Next, we click on the drawing window to make it active and in the layers pane we check the box to show the Mexico Layout "layer" within the drawing window.
This displays within the drawing the size and position of the layout elements currently in use for this drawing. Because the layout element is currently set to *entire component*, it shows that all of Mexico is displayed within the layout element. We can right click on the hatched portion of the layout preview rectangle and choose **Properties** to launch the layout **Properties** dialog.

In the **Properties** dialog we choose **locked rectangle** for **Scope** and press **OK**.

![Properties dialog screenshot](image-url)
In the drawing, a black rectangle appears to show the extent of the locked region that will be displayed in the layout.

We can click and drag this black rectangle with the mouse to change its shape. [Unlike many such re-sizeable items in Manifold, the rectangle does not use edit handles but it still is re-sizable.]

Note that as the black rectangle is resized the contents of the layout element automatically change to show only what is inside the black rectangle. We are using the drawing window to control what is visible in the corresponding layout element in the layout window.

If we resize the rectangle to show only Durango province, that is what is seen in the layout.
Upon close examination it will be clear that the layout window shows a slightly larger region to the left and right of the Durango province than is indicated by the square black rectangle in the drawing window. The reason is that the element in the layout is a more elongated rectangle and so it shows more than the square black box indicated in the drawing window.

When using the locked rectangle option the system will make a best effort to match the display indicated in the locked rectangle in the drawing to the aspect ratio of the element rectangle; however, it could be that great differences in the aspect ratio will allow somewhat more of the region of interest to be displayed.

To see this effect, CTRL-ALT click onto the layout element to select it for editing and resize it so that it is an even wider rectangle with reduced height. Note that the system still positions Durango within the rectangle so that at least the height of both locked rectangle in the drawing and height of what is displayed in the layout are the same.

However, because the rectangle is much wider in aspect than the indicated locked rectangle there will be more of the regions to the East and West of Durango that are visible in the layout.

Note that the hatched area in the drawing shows a preview of the extent of the layout element compared to the black box of the locked rectangle. As the layout element is resized in the layout window, the hatched preview of the layout element in the drawing window will change.

We can deselect the layout element to see clearly the relationship between the extent of the layout element in the layout window and the previews of the layout element and the locked rectangle in the drawing window.

Note: The Edit buttons that appear when a scope of locked rectangle or locked center / scale is used allow specification of the dimensions of the locked rectangle or center / scale view numerically. The scale parameter given is an internal scale parameter expressed in native coordinate system units per point. Normally, the Edit dialogs are not used in favor of setting a locked rectangle visually within the component and setting a locked center and scale by choosing an existing view or object set.


Layer

The layer option for scope in layout elements is similar to the entire component option, except that only that part of the component needed to show the given layer would be shown.

Suppose we create a map, called Mexico Map using the Mexico drawing used earlier. In addition, we have added another drawing as a layer in the map, called Stations. This drawing shows a group of points indicating environmental monitoring stations within a particular region in Mexico. The points are shown using a yellow square box style, with a point size of 4. To continue the style of illustrations used in this topic, the Mexico Map window is shown above approximately the same size as the layout window.

Because the layout still has only one element, the element created from the Mexico Drawing component, we need to add a second element that shows the map. To do so, we drag and drop the Mexico Map component from the project pane into the layout window. We can then resize the new element to whatever convenient size we like.
When a new element is dropped into a layout by default the "white space" it contains is drawn with transparent color. To make the layout more clear, we have right clicked onto the new map element and within the Properties pane have set the element to use white background color and to be drawn with a border.

By default the new element uses entire component as the Scope, so that the entire Mexico Map component is shown. To use only one layer, we first right click on the element and choose Properties and then in the Properties dialog we change the Scope to layer. We choose Stations as the layer to use for the scope.
The element immediately changes so that it shows only that part of the Mexico Map component that is required to show all objects in the Stations layer. New users may be surprised to see that the points in the Stations layer appear much smaller than they do in the Mexico Map window.

They do so because the display in the layout window is a true "WYSIWYG" (What You See is What You Get) display that also functions as a print preview. Because it shows a full sheet of paper zoomed out so that it all fits within a small window, the rather small point icons used (only four printer's points in size) will appear quite small.

If we zoom into the layout using the Views - Zoom To command to a scale of 1:1 we can see that when printed full size the yellow points will be exactly the same size, four printer's points, on the full-sized printed sheet that they are in the map window on the display monitor.

**Printing to Scale**

As can be seen above, printing to scale in Manifold is easy. We simply create a view of the desired scale and then use it as the scope of a layout element.

**To print a drawing to scale:**

1. Open the drawing and pan and zoom so that the desired area of interest is centered in the drawing window. Make sure the drawing is a projected drawing.
2. Use the Views - Zoom to command to zoom to exactly the scale desired.
3. In the views pane, create a new view.
4. In the project pane, right click onto the drawing and choose Create - Layout to create a layout using that drawing.
5. Open the layout and right click on the element and choose Properties.
6. In the Properties pane, for the Scope use the view created above.
7. If desired, resize the element in the layout to show more or less of the view at the given scale.

New users are at times confused by the relationship between what is shown in a layout viewport and a desired scale. If we decide to show a map at a desired scale, such as 1 to 10 million, then the only way to control how much of it is shown is to vary the size of the element in which it appears. A larger element will allow more of the map to be shown.

Conversely, if we want a particular region to be shown with no more or less of the map being visible than that particular region, then we cannot constrain the display to a fixed scale. In this case the view shown by the element will have to be zoomed in or zoomed out (re-scaled) so that only the region desired is seen within the element.

See Also

Layouts
**Multipage Layouts**

Manifold can print a layout using more than one page. This capability is used in two ways:

- To print a layout using many pages that are then pasted or tiled together to create a single image at a larger size than a printer can ordinarily print.
- To create a “map book” that shows a layout in the form of multiple pages that are formed into a book.

In both cases, the **Layout - Pages** dialog is first used to specify the number of pages (that is, the number of sheets of paper) to be used.

**To create a multipage layout:**

1. Create a layout and open it.
2. Choose **Layout - Pages** to open the **Pages** dialog.
3. In the **Pages** dialog specify the number of pages in the horizontal X direction and the number of pages in the vertical Y direction. Press **OK**.
4. Drag and drop the component to be spread across the pages into the layout.

**Very important:** Multipage layouts in current Manifold releases are strongly page oriented, where the boundary of a paper sheet is an absolute boundary. Components can be used in layout elements either as tiled continuously across all pages or as appearing within individual, repeated layout elements on every page. However, layout elements cannot appear on only one or two pages of a multipage layout, partially on one page and partially on another, as if the multipage layout represented a single large sheet of paper.

**Example**

Suppose we have a map that shows Europe with major cities.

![Map of Europe](image)

We can create a layout showing the map as seen above. **Page Setup** has been used to show the page in letter sized paper in **Landscape** orientation. To use more than one paper sheet to print the layout, choose **Layout - Pages** to open the **Pages** dialog.
In the **Pages** dialog we enter the above values to create a layout that has three pages horizontally and three pages vertically.

If we right clicked onto the map element in the layout and chose **Properties** we would see that the **Scope** is set to **entire component** and the **Paging** option is set to **continuous**. This means that the entire component is resized to fit to the maximum size possible within the pages available for the layout.

If we were to print the layout, six sheets of paper would be printed. Each paper sheet will have the correct section of the map printed so they can be tiled together to form a much larger printout. We could cut the pages along their inner margins (the region where the printer cannot print) and glue them together in a mosaic that is three sheets by two sheets to form a single, large printout.
If we right click on the map element, choose Properties and then change the Paging option to individual. This changes the usage of the component within the layout so that it is repeated on each page. The main reason we use this paging option is to place a repeated element on each page of a series of maps, such as those in a map book. The element might be a logo, a copyright notice, or a text expression such as Page number [Page] that gives the page number of each page.

**Multipage Layouts and Views**

When a view is used as the Scope for a layout element the component will be shown in the layout at the same scale as the view. Recall that we create a view by opening a component window such as a map window, drawing window, image window or surface component and then panning and zooming until the view seen in the window is the view we want. We then save it in the Views pane.

If the component is projected, saving a view at a given zoom level is the same thing as saving a view at a given scale. If the Adjust display scale for monitor resolution option is set in the Tools - Options dialog, the view we see at the zoom/scale in use will be literally the same size as what will be printed out if that view is used in a layout.

Suppose in the Europe example map seen above we have created a view called Central that is a zoomed-in view of the central part of Europe.
Suppose we create another layout, also in Landscape paper orientation, using the Europe map. If we right click on the map element in the layout, choose Properties and then set the Scope to View and choose the Central view we might see a display like that above.

The view is displayed at the same scale that was used to create it. The command to use a view for the scope of a layout element is a command to Manifold to show as much of the view at the specified scale that will fit within the size of the layout element. In the illustration shown above the layout element fills the entire page so as much of the view as can fit at the given scale will be printed.

Now let's try something interesting: we will use the Layout - Pages dialog to change the layout to a multipage layout that is three pages by two pages in size.

The layout expands to three pages by two pages in size, but now instead of showing just the central part of Europe the layout shows almost all of Europe. Although this may seem not to be the right behavior at first glance, in fact it is the correct function because of what we have told Manifold to do.
Because we are using a View as the scope, the element is printed at the scale used for the view. Using the same scale means that the sizes of the countries are printed at a fixed size on the paper, at whatever is the right size for the scale being used. If we make the paper sheet larger, either by printing on a larger piece of paper or by having a multipage layout, the sizes of the countries will still be the same. Using a larger piece of paper simply means there is more space available to print a larger portion of the view.

In the example above, if we look at, say, the middle page in the lower row we can see that the countries are printed at exactly the same scale as they were in the preceding illustration where only one sheet was used. The view is centered upon the multipage layout and it extends across the multiple sheets as far as the map extends at that scale.

We can confirm this by changing the Paging option from continuous to individual, which results in the display above. In this case the view is still used, but it is repeated on each page. If we compare the size of any country in this display to the size of the country when one sheet was used in the layout or when continuous paging was used we can see that in all three cases the size of the country is the same. That happens because we are using a view and views are always printed at the scale used at the view.

Suppose our objective is to show just the central part of Europe, but spread across a page layout that is three pages by two pages in size. How should we accomplish that?
The solution is to open the map component, open the Views pane and create another, more zoomed in view, perhaps called Central2. We can then use that view in the print layout to create a scene like that seen above.

"Map Book" Resizing Behavior

A map book is a large map that has been tiled into individual sheets that can then be bound as a book. Many consumer road atlases are produced in this form so drivers need not fold and refold large, unwieldy map sheets. The continuous paging mode of a component in a multipage layout can be used to produce a map book. If margins are set to allow room for a binding on one side the individual sheets will print correctly and can be bound into a book.

At times we would like to resize the element in the layout so that it does not quite fill each available sheet. We can do this to leave some empty space on each sheet for inclusion of other graphic elements. Let's begin with the zoomed in Europe example from the previous illustration.

We click on the map element to select it. It fills the layout.
If we **CTRL-ALT-click** on the map element in one of the pages of the layout, the map element will be selected for editing. We can resize the region in which the element appears by an editing handle. For example, we can drag the editing handle in the lower right corner to make the region smaller.

The result is that all of the regions for this element on all of the pages of the multipage layout are resized at the same time. This happens because in **continuous** paging mode the layout element for this map extends across all pages within the region on each page it is allowed to occupy. We can click outside of the element to turn off selection. Any alterations to this selection frame will apply to the element on all pages. For example, when we resized it the "resize" operation was applied to the element on all pages.
The result at first is slightly baffling. It appears to show a map cut into portions. However, when we examine the
map more closely we can see that after allowing for the margins each portion fits against adjacent portions of the
map exactly.

This is the behavior we would like in a map book. Suppose we printed this layout and numbered the resulting
pages as numbers 1, 2, and 3 across the top row and then 4, 5 and 6 in the bottom row and then assembled the
pages into a book. If we began reading the book with the first page (showing part of England) it is clear that if we
wanted to continue in the map book into France we would go to page 4 and if we wanted to continue East we
would go to page 2.

Text Components in Multipage Layouts

A text component is a component such as a comments or script component that consists of text. Text
components have no properties page since they are always printed as continuous text through whatever
element boxes they appear in within the multipage layout.

Text Elements in Multipage Layouts

We can make it easier for map book users by adding a page number to each page using a text element.
In the layout, we zoom into the lower right corner of the first sheet, and then use the Insert Text tool to draw a box for a text element.

In the Insert Text dialog we construct a text line consisting of

Page [Page X], [Page Y]

using the built-in [Page X] and [Page Y] expressions. These give us the page number in the column and row of the page layout. Press OK.
The result is a page number printout that will be different on each page that is printed from this layout.

If we zoom into the page number that is on the first page in the bottom row we see that it is given as Page 1, 2. We can edit the appearance of a text element by CTRL-ALT clicking it and then using the format toolbar to change the font. Changing any of the text elements will change all of them on all of the pages.

Some users might not like the "grid" style of page numbering used. We can always put a simple page number on the page by double-clicking the text element and changing the text line to

Tile [Page X], [Page Y] on Page [Page]
This results in a text element like that seen above, shown on the middle page of the bottom row.

**Citing Adjacent Pages / Page Filters**

In a map book it is often convenient to place notations at the margin of a page stating on which page of the book the adjacent map sheet may be found.

Suppose we start with a multipage layout showing Europe in **continuous** paging mode where the Europe element has been resized to allow some additional white space on the edges of each map.

Using the **Insert Text** tool we can draw a text box.
When the Insert Text dialog opens we construct a text string consisting of To Page [Page Below]. The [Page Below] expression will be replaced automatically with the page number of the sheet that appears below this sheet in the multipage layout.

The result (with greatly enlarged font size so the text is readable in this small illustration) is that each page contains a text element that includes the page number of the sheet below it.
Using the Insert Text tool we can add a page number to the upper right corner of each sheet. The text string used is simply [Page] to provide the page number. Note how the text caption at the bottom of the map in the upper left corner sheet reads "To Page 4" and that the sheet adjacent (the continuation of France to the South) is indeed page 4.

Note also that since there are no sheets adjacent to the bottom edge of the bottom row of pages, the captions there all read "To Page none". To eliminate these we can use the Page filter property to specify which pages will be printed.

Right click on one of the text elements and choose Properties from the context menu. In the Properties dialog enter 1-3 for the Page Filter. This specifies the pages on which this text element will appear. The Properties dialog for a text element also allows choosing a border for the text box.
The result is a much cleaner display.

In a similar way, we could add a text element to the upper margin using the string To Page [Page Above] to automatically print the page adjacent to the top margin. In the illustration above we have used the Properties - Page Filter option to specify the upper text element is to be printed only on pages 4-6.
With a bit of tinkering we can add yet more text elements using the [Page Left] and [Page Right] expressions to automatically note which pages appear to the left and right of a given sheet.

Each arrow is a separate text element. They were created by using the Wingdings 3 Windows font, which prints a left arrow for a lower case "f" character and a right arrow for a lower case "g" character. To create a left arrow, insert a text element that contains "f". Next, select the text element for editing by CTRL-ALT clicking on it. In the format toolbar, change the font to Wingdings 3 (if we have it available on our system) and then change the size of the font to make the arrow the size desired.

Next to each arrow is a text element containing [Page Left] or [Page Right]. The page numbers in the upper right corner have been edited (by double clicking them) to contain Page [Page].

There are, of course, many possible ways of using Manifold’s text tools to create notes on adjacent pages. The illustrations shown here were designed mainly to show the method within the very small illustrations available within a Help file and could obviously be improved in a full size display.

**Borders and Multipage Layouts**

The Border each page option found in the Properties of a layout element controls how borders are printed for elements that use continuous style paging in a multipage layout. The option is enabled when using a border.
When the **Border each page** option is checked, Manifold draws a border for each individual page as would be used for map books. When the option is not checked, Manifold draws a single continuous border as would be used when the multipage printout is intended to be tiled together into a single large print.

**Printing Multipage Layouts**

Multipage layouts are printed just like any other layout with **File - Print**. However, if desired, the **Page Range** section of the **Print** dialog can be used to specify that only some pages of the layout are printed.

**Notes**

Some multipage layouts of large maps using white backgrounds may result in some pages with no printing on them because the page falls entirely within a "whitespace" area of the map. By default, all pages are printed even blank pages. To save paper, we can uncheck **Print blank pages in multipage printouts** in **Tools - Options - User Interface**.

Using fonts such as **Wingdings 3** to add graphics with text elements is fun, but not particularly portable. If we move the `.map` project to a different system that has not had **Wingdings 3** installed on it and we open the layout, the arrows will not appear. Instead, the "f" and "g" characters will probably appear in the best default font that the system has available.

On the other hand, portability will be retained when saving a layout to `.pdf` because Manifold writes fonts to such files as "curves" or "outline" fonts so there is no need to have the font available locally when the `.pdf` is viewed.
Legends, Scale Bars and North Arrows in Layouts

A print layout in Manifold can include more than one component, that is, more than one drawing, image, surface or map. Each component can have its own legend, scale bar and North arrow specified. As many legends, scale bars and North arrows as desired may be added, with each one controlled by one of the components in the layout.

Adding legends, scale bars and North arrows to layouts in Manifold is therefore a two step process:

- Open the component and configure the legend, scale bar or North arrow as desired.
- Add the component to the layout, right click on the component, choose Properties and set the desired display mode for the legend, scale bar or North arrow. The default mode is auto, which means that if the legend, scale bar or North arrow is visible in the component window it will be visible in the layout.

Alternately, the legend, scale bar or North arrow may be added to the layout first and then configured in its parent component:

- Open the print layout and use Insert Legend, Insert Scale Bar or Insert North Arrow to insert the legend, scale bar or North arrow desired. If there is only one component element in the layout the legend, scale bar or North arrow will automatically be bound to that component. If there is more than one component element in the layout, a dialog will pop up asking to which component element the legend, scale bar or North arrow should be bound. In the pop up dialog, specify the component that will control the legend, scale bar or North arrow.
- To configure the legend, scale bar or North arrow, right click it and choose Edit, or double click on it to open it for editing. This will pop open the standard configuration dialog that would appear in View - Edit, View - Scale Bar or View - North Arrow.

To add a legend to a print layout (Method 1)

1. Open the component to be added to the print layout.
2. Choose View - Legend and configure the legend as desired.
3. Open the print layout.
4. Drag and drop the component from the project pane into the layout.
5. Right click on the component in the layout, choose Properties and, if the legend is not visible, change the Legend setting to show.

To add a legend to a print layout (Method 2)

1. Assuming the layout already has a component that can control a legend, use the Insert Legend tool to draw a box where the legend is to appear.
2. In the pop-up dialog choose the name of the component that will control the legend.
3. Right click on the legend in the layout and choose Edit, or double-click on the legend to open it for editing.
4. Configure the legend as desired.

The main difference between the first method and the second method is that in the first method the legend is a part of the layout element that controls it, and it may not be moved beyond that layout element. In the second method, the legend is created as an independent layout element and may be moved anywhere in the layout.

The procedure for adding scale bars or North arrows is similar. For example, we can adjust the configuration (style, options, etc.) of the legend, scale bar or North arrow working in the component window. Next, we can use the second method above, using the Insert Legend tool, for example, to add the item to the layout and choose the component to control it. The legend, scale bar or North arrow will have the same appearance as configured in the component window.

Note that every component that can host a legend, scale bar or North arrow can have only one configuration for that legend, scale bar or North arrow. It doesn't matter if, for example, a drawing is opened in its own window and the legend configured as desired, or if a legend for that drawing is right-clicked in a layout and the Edit command is used to configure the legend. In both cases the legend will take on the same appearance whether it is in the print layout or in the drawing window.
**Properties Options**

Whether or not accessories such as a legend, scale bar or North arrow appear in a layout element depend on the Properties settings for that element. Each accessory can be configured to auto, hide or show as follows:

- **auto**: If the accessory is turned on (show) in the main component window it will appear in the layout. If it is turned off (hide) in the main component window it will not appear in the layout.
- **hide**: Always hide the accessory in the layout even if it is turned on in the main component window.
- **show**: Always show the accessory in the layout even if it is turned off in the main component window.

**Changing Legends, Scale Bars or North Arrows**

To delete a legend, scale bar or North arrow from a layout, right click on the element and in the Properties dialog change the accessory's display mode to hide. If the legend, scale bar or North arrow was inserted using a tool like Insert Legend, select it for editing by CTRL-ALT clicking it and press the Delete key.

To edit the appearance of a legend, scale bar or North arrow, right click on it and choose Edit, or simply double click on it to open it for editing.

Note that accessories may be configured either in their host component's window or in the print layout. For example, if we would like to change the style of a legend that has been added to a map element in a layout, we can open the map in its own window and change the style of the legend there, or we can edit the legend in the layout. In either case, any change made in the map window will appear in the legend in the layout window and vice versa. This is true of scale bars and North arrow changes as well.

**Example**

See the Adding a Legend example topic for a step by step example of configuring a legend and then adding it to a print layout.
Printing Tables

Tables may be printed using three methods in Manifold:

- Open a table window and choose File - Print.
- Create a layout from the table and then print the layout.
- Create a layout using other components, drag and drop the table into the layout and then print the layout.

Which method is used depends upon the purpose of the layout. Tables are often printed by themselves to provide a printed record of the data they contain. In that case it is usually easiest to create the layout from the table. Tables are often also used as elements within a layout that may contain maps or drawings to provide data attributes or other information related to the objects shown in the maps or drawings. When the layout has already been created from some other element the table can be dragged and dropped into the layout and then configured as desired.

Creating a Layout from a Table

The simplest way to print a table from a layout is to create a layout using the table to be printed.

Creating a print layout from a table:

1. Open the table.
2. Hide any undesired columns.
3. Right click onto the table in the project pane and choose Create - Layout.
4. Open the layout and configure the table within that layout as desired. CTRL-ALT click onto the table to edit it. Right click onto the table and choose Properties to configure the tables' properties, such as which columns are displayed.
5. Tables may be formatted (font, foreground and background color) by selecting the table element for editing with a CTRL-ALT click and then using the Format toolbar.

When a layout is created from a table it will automatically be created as a multipage layout, if necessary, to print the entire table. If all columns and records fit on one page of a layout the table will be automatically centered on the layout page.

See the Create a Print Layout from a Table example topic for a step by step example.

Adding a Table to an Existing Layout

Another way to print a table is to add it to an existing layout by dragging and dropping it into the layout. When dragging and dropping a component into an existing layout, Manifold will drop the component into the layout using an element that does not fill the entire page. This allows dropping a new component into a layout without obscuring any other components that might already be there.
For example, if we have an existing layout created using a drawing we can drag and drop a table into that drawing. The table will appear as a new layout element at the point it was dropped.

We can `CTRL-ALT` click on the table to select it for editing, which will enable us to resize the table element to allow more of the table to be visible. This particular table illustrated is the table from the `mexico_eg.mif` drawing on the Manifold CD.
To resize the table we can drag the upper left editing handle.

When the table is resized more rows and columns will be visible. Note that since table background color is transparent by default the drawing that was already in the layout can be seen through the transparent parts of the table.

**Table Properties in a Layout**

Right click on a table element in a layout and choose **Properties** to set the following options:

- **Scope**  
  Choose **Entire component, saved selection, selection, or view** to specify the contents of the table. If views or selections have been saved, choosing **view** or **saved selection** will enable the choice box to the right with all of the views or saved selections available. The **Scope** setting allows us to print the entire table or just a selection within the table.

- **Paging**  
  Controls the printing of this table within multipage layouts:
  - **continuous** - Spread output evenly across the entire range of pages they appear on.
  - **individual** - Confin output to only that portion of a page on which the component appears and replicate this output.
likewise on all pages. Used to create overview maps with a printout repeating on all pages.

**pages or page ranges**

Used with multipage layouts to control which page this element appears in. Enter a series of page numbers or page ranges separated by commas. White space is ignored and reverse ranges are interpreted correctly. For example, We could enter 1, 2, 5-8 or 1, 2, 8-5 and get the same results. See the Multipage Layouts topic.

**Columns**

Choose auto or custom. The auto setting will automatically show or hide columns in the table element in the layout as columns are shown or hidden in the table window. The custom setting enables the columns pane in the dialog to allow choice of columns by checking them.

**[Columns pane]**

Enabled when the Columns option is set to custom. When enabled, the background of the columns pane will change to white and the toolbar will be enabled.

**[Toolbar buttons]**

The column pane's toolbars are enabled when the Columns option is set to custom.

- **Select All** - Check all boxes to print all columns in the table.
- **Select Inverse** - Uncheck all checked columns and check all unchecked columns. A fast way to show all but one column: click Select None, check the one column not desired and then click Show Inverse.
- **Select None** - Uncheck all columns in the table so none will be printed.
- **Move to Top** - Move highlighted column to the top of the list. This places the column at the leftmost position of the table.
- **Move Up** - Move highlighted column up one position in the list. This moves the column one step to the left in the table.
- **Move Down** - Move highlighted column down one position in the list. This moves the column one step to the right in the table.
- **Move to Bottom** - Move highlighted column to the bottom of the list. This places the column at the rightmost position of the table.

**Border**

Show a border for the table. By default, no border is shown. thick or thin borders may be selected.

**Column headers**

Display the names of table columns in boldfaced column headings.

**Grid lines**

Draw grid lines to separate rows and columns in the table.

**Shade alternate rows**

Use a shaded background for alternate table rows.

When adding a table to a layout, by default the auto setting for the Columns property will cause the table to be printed in the layout showing the columns that were displayed the last time the table window was opened. Any columns that were hidden from display will not be printed. This may be changed in two ways:

- Open the table window and show or hide columns using the View - Columns dialog or by right clicking onto column heads in the table and choosing Hide.
- Right click onto the table, choose Properties and change the Columns setting to custom. Next, check and uncheck the columns as desired in the columns pane.
Changing Colors and Fonts

To change the foreground (text) and background color of a table or to change the font used, select the table element for editing by CTRL-ALT clicking it and then use the Format toolbar to change foreground and background color and font.

See Also

Create a Print Layout from a Table
Import and Export

Import and Export

Drawings, images and terrains may be imported into a Manifold project from a very wide range of formats using menus under File - Import. Different formats and variations thereof amount to over one hundred different types of files that may be read by Manifold.

Manifold organizes importers by type of component. To import a file, choose File - Import and then the type of component desired in the sub-menu. For example, in the image below we are about to import an image.

In the Import dialog that follows choose the appropriate file type desired from the pull-down list available in the Files of type list box.

For example, in the screen shot above we have chosen JPEG file types. Note that some importers use the same file type choice for filenames that have similar, but different extensions. For example, the same JPEG file choice importer is used for image files that have .jpg as well as .jpeg extensions.

The Files of type choice acts as a filter for what is displayed in the browse pane in the Import dialog. Simple file types are imported directly from the Import dialog by choosing the desired type in the Files of type box. More complex imports from data sources such as databases are imported by choosing Data Sources () in the Files of type box and then using the Data Source dialog.

When importing a simple file type, browse over to the file desired and click on it and press Open to import it, or simply double click on the file in the browse pane to open it and commence the import. We can import several files at once by using CTRL-click to choose more than one file when browsing in the Import dialog. We can also click on one file and then SHIFT-click on another to highlight all the files in between if we wish to choose several files for import.

Some importers are designed for use with files that are saved from specific applications. For example, the MapInfo MIF importer is used to import MapInfo files saved to .mif/.mid format. This format uses two files, a .mif file and an accompanying .mid file, to save a GIS drawing. To use this importer we need choose only the .mif file.

A few importers, such as the VMAP and the TIGER/Line importers are designed to work with very complex formats that provide information in many layers or drawings within one or more files. VMAP for example, is an extremely complex format involving numerous subfiles that are indexed by a main DHT file. Such formats are normally imported as numerous drawings at one time and automatically organized into one or more maps within the project.
Some formats, such as .e00 may contain drawings, surfaces or images. .e00 importers therefore appear as choices within the import dialogs for drawings, surfaces and images.

**SDTS** format may contain either drawings or surfaces. Import both types of components using the **File - Import - Drawing** choice even if you know the SDTS contains a "DEM" surface. A surface will automatically be created if the SDTS contains terrain elevation DEM data.

Not all importers are documented within this Help file. Many importers (especially for images) have no options and are operated by simply choosing the file to be imported. Other importers import data from very complex formats that require documentation beyond the scope of this document. Please consult the originator of the format for documentation on features.

**Strings are Trimmed on Import**

An option set in **Tools - Options** by default trims leading and trailing white space from strings. This may be turned off if desired. The trim function trims white space so that strings like " a b cdef " (with spaces before the a and spaces after the f) are trimmed to "a b cdef". Internal whitespace is unaffected.

**Text Components Import / Export as ANSI Text**

Manifold supports imports and exports of text components such as comments, scripts and queries. All such imports and exports are accomplished using ANSI text, not Unicode.

Text components may be imported or exported using any file extension. For example, .hdr "header" files are often plain ASCII text files that provide information about other files being imported. These may be imported as text to create a comments component.

**Exporting Maps**

A map component may be exported as a .dxf file, in which case all drawing and label layers will be exported into .dxf layers. Maps may also be exported to **XML** or **KMZ** format.

Otherwise, the layers that form a map should be exported into appropriate formats, for example, images exported into image formats, surfaces into surface formats, drawings into GIS or CAD drawing formats and so on.

**XML Files Created upon Export and Used on Import**

Not all formats to which Manifold can export have the ability to correctly save projection information. As a safety measure to ensure that projection information is never lost, Manifold always writes an accompanying .xml file when exporting drawings, images, labels, maps or surface components to a file. The accompanying .xml file contains coordinate system information and some other metadata. The XML file is created for all formats, even those that correctly save coordinate system information.

When Manifold imports drawings, images, labels or surfaces, the system will check for an accompanying .xml file that might have been written by Manifold. If such an .xml file exists, Manifold will read it and use the information it contains to load a correct coordinate system.

See the XML Accessory File Format topic for details on the internal organization of this file.

**Important Note when Using 64-bit Manifold Editions**

Due to a lack of required Microsoft facilities in 64-bit Windows systems, Manifold in 64-bit mode cannot export, import, export or link to **DB**, **HTML**, **MDB**, **XLS** or **WKx** format files. This includes no access to the **MDB** parts of Manifold **MFD** and MapInfo **TAB** imports. The workaround for importing or exporting such files is to launch Manifold in 32-bit mode by using the **Manifold System (32-bit)** shortcut, perform the export from or import into a .map project file and then re-launch Manifold in 64-bit mode using the **Manifold System (64-bit)** shortcut.

Linking is more complex: the data must be in some format usable in 64-bit mode within 64-bit Windows systems, such as a SQL Server database. Alternatively, the data can be kept within a Manifold .map project file and linked using the Manifold ODBC driver.

**Tech Tips**
At times we would like to fetch a component that's been saved in a .map project file for use in another project file. One way to do this is to open a second instance of Manifold and to open the saved .map project. We can then **Copy** components from that project file, switch to our current working instance of Manifold and **Paste** them into our current project. Another way is to use the File - Import - Component command to import a component directly from a .map project file.

Whenever importing projected data (drawings, images or tables) from formats that do not provide projection information, be ready to visit the Edit - Assign Projection dialog after the import to make any necessary adjustments. See the Import a Shapefile example topic for a simple example and the Import Drawing - SHP, Shapefiles topic for a more extensive example of using Edit - Assign Projection to adjust projection parameters after import. See Creating Drawings from Geocoded Tables for information on table imports and usage when the table contains geocoded projected or unprojected data.

**See Also**

Importing Drawings  
Importing Images  
Importing Surfaces  
Importing Tables  
Linking Tables  
Exporting Drawings  
Exporting Images  
Exporting Layouts  
Exporting Surfaces  
Exporting Tables
XML Accessory File Format

Not all formats to which Manifold can export have the ability to correctly save projection information. As a safety measure to ensure that projection information is never lost, Manifold always writes an accompanying .xml file when exporting drawings, images, labels, maps or surface components to a file. The accompanying .xml file contains coordinate system information and some other metadata. The XML file is created for all formats, even those that correctly save coordinate system information.

When Manifold imports drawings, images, labels or surfaces, the system will check for an accompanying .xml file that might have been written by Manifold. If such an .xml file exists, Manifold will read it and use the information it contains to load a correct coordinate system.

Example

GeoTiffs don’t require an .xml file. A “GeoTiff” is simply a .tif file with internal tags that save projection information. However, even when writing a GeoTiff Manifold will also create an accompanying .xml file. The .xml file is created by Manifold for all exports, even those that don’t need it, in something of a “belt and suspenders” move to save people from inadvertently losing projection info.

Even in the case of a GeoTiff the creation of a redundant .xml file is a valuable idea. Consider the following scenario: we georegister an image and save it as a GeoTiff. So far, so good. Now, let’s say we edit the .tif using PhotoShop, which for all of its image editing prowess doesn’t understand the GeoTiff tags embedded in the .tif and won’t preserve them. After editing in PhotoShop the image is now a plain, ordinary .tif that’s no longer georegistered and has lost all projection info.

However, as long as we haven’t changed the size or rotation of the image or renamed the file, we can still open it in Manifold and Manifold will automatically recover the projection information from the auxiliary .xml file thoughtfully written by Manifold when we first exported the .tif. If PhotoShop had not erased the GeoTiff tags inside the .tif we wouldn’t have needed the .xml file, but clearly it is very useful to have the .xml on hand to recover from what otherwise might have been a disaster caused by the PhotoShop editing session.

XML File Format

This section provides details on the internal organization of the accessory .xml file created on export and read, if available, on import.

Note: in the tables listing tags below, such as those that may be used for parameters, it is understood that a <tag> is matched with a </tag>. Instead of writing <tag> ... </tag> in the tables to indicate a <tag> followed by some intermediate material and then closed with a </tag>, we simply write “<tag>”

XML File Format:

<?xml version="1.0" encoding="UTF-8"?>
<data>
  <coordinateSystem>
    ...  <!-- coordinate system data
  </coordinateSystem>
  ...  <!-- other data
</data>

Coordinate System Data Format:

<coordinateSystem>
  ...  <!-- parameters in the form of <par>...</par> tags
</coordinateSystem>

Parameters:
Tags used with various coordinate systems:

<system>Albers Conical Equal Area</system>

  <centerLat> center latitude </centerLat>
  <centerLon> center longitude </centerLon>
  <firstStdLat> 1st standard latitude </firstStdLat>
  <secondStdLat> 2nd standard latitude </secondStdLat>

<system>Azimuthal Equidistant</system>

  <centerLat> center latitude </centerLat>
  <centerLon> center longitude </centerLon>

<system>Bipolar Oblique Conformal Conic</system>

  (none)
<system>Bonne</system>
<centerLat> center latitude
<centerLon> center longitude

<system>Cassini</system>
<centerLat> center latitude
<centerLon> center longitude

<system>Cylindrical Equal Area</system>
<centerLat> center latitude
<centerLon> center longitude

<system>Cylindrical Equidistant</system>
<centerLat> center latitude
<centerLon> center longitude

<system>Double Stereographic</system>
<centerLat> center latitude
<centerLon> center longitude

<system>Eckert IV</system>
<centerLon> center longitude

<system>Eckert VI</system>
<centerLon> center longitude

<system>Equidistant Conic</system>
<centerLat> center latitude
<centerLon> center longitude
<firstStdLat> 1st standard latitude
<secondStdLat> 2nd standard latitude
<system>Gnomonic</system>

- <centerLat> center latitude
- <centerLon> center longitude

<system>Goode's Homolosine (Interrupted)</system>

- (none)

<system>Hammer</system>

- <centerLon> center longitude

<system>Krovak Oblique Conformal Conic</system>

- <centerLat> center latitude
- <centerLon> center longitude
- <centerLineAzimuth> azimuth of the center line
- <pseudoStdLat> pseudo-standard latitude

<system>Lambert Azimuthal Equal Area</system>

- <centerLat> center latitude
- <centerLon> center longitude

<system>Lambert Conformal Conic</system>

- <centerLat> center latitude
- <centerLon> center longitude
- <firstStdLat> 1st standard latitude
- <secondStdLat> 2nd standard latitude

<system>Lambert Conformal Conic (Single parallel)</system>

- <centerLat> center latitude
- <centerLon> center longitude

<system>Latitude / Longitude</system>
<system>Mercator</system>
  <centerLat>center latitude</centerLat>
  <centerLon>center longitude</centerLon>

<system>Miller Cylindrical</system>
  <centerLon>center longitude</centerLon>

<system>Modified Polyconic</system>
  <centerLon>center longitude</centerLon>
  <firstStdLat>1st standard latitude</firstStdLat>
  <secondStdLat>2nd standard latitude</secondStdLat>
  <trueScaleLon>longitude of true scale</trueScaleLon>

<system>Modified Stereographic</system>
  <centerLat>center latitude</centerLat>
  <centerLon>center longitude</centerLon>

<system>Modified Transverse Mercator</system>
  (none)

<system>Mollweide</system>
  <centerLon>center longitude</centerLon>

<system>Mollweide Interrupted</system>
  (none)

<system>Oblated Equal Area</system>
  <centerLat>center latitude</centerLat>
  <centerLon>center longitude</centerLon>
  <ovalRotationAngle>rotation factor</ovalRotationAngle>
<ovalShapeM> M shape factor
<ovalShapeN> N shape factor

<system>Oblique Mercator (A)</system>
<centerLat> center latitude
<firstLat> 1st point latitude
<firstLon> 1st point longitude
<secondLat> 2nd point latitude
<secondLon> 2nd point longitude

<system>Oblique Mercator (A, centered)</system>
<centerLat> center latitude
<firstLat> 1st point latitude
<firstLon> 1st point longitude
<secondLat> 2nd point latitude
<secondLon> 2nd point longitude

<system>Oblique Mercator (B)</system>
<centerLat> center latitude
<centerLon> center longitude
<centerLineAzimuth> azimuth of the center line
<rectifiedGridAngle> rotation factor

<system>Oblique Mercator (B, centered)</system>
<centerLat> center latitude
<centerLon> center longitude
<centerLineAzimuth> azimuth of the center line
<rectifiedGridAngle> rotation factor

<system>Orthographic</system>
<centerLat> center latitude
<centerLon> center longitude
<system>Polar Stereographic</system>

- <centerLon> center longitude
- <trueScaleLat> latitude of true scale

<system>Polyconic</system>

- <centerLat> center latitude
- <centerLon> center longitude

<system>Robinson</system>

- <centerLon> center longitude

<system>Sinusoidal</system>

- <centerLon> center longitude

<system>Space Oblique Mercator</system>

- <ascendingNodeLon> longitude of the ascending node
- <completedOrbits> number of completed orbits
- <orbitInclinationAngle> orbit inclination angle
- <rotationToAscendingNode> rotation of the Earth to the ascending node
- <satelliteRevolutionTime> satellite revolution time

<system>Space Oblique Mercator for Landsat</system>

- <orbitNumber> orbit number
- <orbitPathNumber> orbit path number

<system>Stereographic</system>

- <centerLat> center latitude
- <centerLon> center longitude

<system>Tilted Perspective</system>

- <centerHeight> center height
<centerLat> center latitude
<centerLon> center longitude
<tiltAngle> tilt angle
<tiltAzimuth> tilt azimuth

<system>Transverse Mercator</system>
<centerLat> center latitude
<centerLon> center longitude

<system>Van der Grinten</system>
<centerLon> center longitude

<system>Vertical Perspective</system>
<centerHeight> center height
<centerLat> center latitude
<centerLon> center longitude

<system>Wagner IV</system>
<centerLon> center longitude

<system>Wagner VII</system>
<centerLon> center longitude

<system>Winkel Tripel</system>
<stdLat> standard latitude

Notes

All latitudes and longitudes are in degrees. All angles are in degrees. All heights are in meters.

The local coordinate system is used as follows:

Suppose we have a SHP file in UTM 13N. Given the xf:yf coordinates stored in the SHP file, xc:yc are the coordinates in UTM 13N:

\[ xc = xf \times \text{localScaleX} + \text{localOffsetX} \]
\[ yc = yf \times \text{localScaleY} + \text{localOffsetY} \]
Alert readers will note the above table does not include a listing of various national grids. National grids are merely customized versions of the standard projections. In an XML file for a national grid, the `<name>` tag will contain the name of the grid (for example, "Michigan ..."), the `<system>` tag will contain the name of the underlying projection (for example, "Lambert Conformal Conic" or "Transverse Mercator"), and all other tags will contain the parameters for the underlying projection.

See the XML files on the Manifold CD in the documentation/xml files folder for examples of XML files that specify various projections. See also the Custom Coordinate System (Projection) Presets topic.
Import Component

Choose File - Import - Component to import components from Manifold .map project files. Choose the .map file to be opened in the resulting Import dialog and then choose the components to be imported in the Import MAP File dialog that follows.

Select All - Check all fields. All data fields will be imported with this drawing.

Select None - Uncheck all fields. No data fields will be imported. Tech tip: Use this to uncheck all fields before checking those desired.

Select Inverse - Uncheck all previously checked fields and check all previously unchecked fields. This is a fast way to check only a few columns out of many: uncheck those desired and then push Select Inverse.

Any component may be imported from a previously saved .map file. Labels components that are bound to drawings will be imported as unbound labels components.

Tech Tip: Save Frequently Used Components in .Map Files or Databases

The .map format used to save Manifold projects automatically compresses the data contained in the .map file using techniques similar to those used by popular "zip" compression utilities. Therefore, there is no need to "zip" a .map file with utilities such as WinZip to save space or to speed up transfers over Internet - the .map file is already compressed.

We can keep collections of frequently used drawings or other components in .map files and then use File - Import - Component to import them from the .map file into the current project whenever desired.

Another way to save frequently used components is to keep them in a spatial DBMS and then import or link the component from the spatial DBMS. This does require installing a DBMS, but it can be a very fast way of retrieving frequently-used components and sharing them with multiple people.

One final way (requiring Enterprise Edition) to save frequently used components is in an Enterprise Server. This can be a fast and convenient way to keep components organized and to share them in different projects.

Import Drawing

Importing Drawings

Choose File - Import - Drawing to import drawings from formats used to save drawings. Such formats might be said to save "maps" in other GIS systems. Drawings may also be linked from database tables. See the Linked Drawings and the Geometry in Tables topics for more information on linking drawings.

The Files of type choice acts as a filter for what is displayed in the browse pane in the Import dialog. Simple file types are imported directly from the Import dialog by choosing the desired type in the Files of type box. More complex imports from data sources such as databases are imported by choosing Data Sources () in the Files of type box and then using the Data Source dialog.

Formats used to save drawings vary widely in their capabilities. Some formats save projection information and some, such as .dxf or many types of .shp files, do not. When a drawing format contains projection information, Manifold will automatically use that projection information.

When a drawing format does not save projection information, for GIS formats such as ESRI .shp files the drawing will be imported into Latitude / Longitude projection as if the units contained were degrees. Most people are sensible enough not to use .shp and similar formats to publish projected maps so this will usually work.

For drawings imported from non-GIS formats such as AutoCAD .dxf, the drawing will be imported using the same default Orthographic projection used for non-geographic images. Such drawings will always require manual adjustment of parameters in the Edit - Assign Projection dialog.
If a drawing format does not save projection information, that information must be added after import via the Edit - Assign Projection dialog.

**Use Folders to Organize Imports**

Some drawing formats, such as those used for VMAP, will result in numerous drawings, tables, text components and maps. It's best to first create a folder in the project and to click on the folder to highlight it before beginning import of such drawings. The resultant mass of components created will then be created within the folder.

**Strings are Trimmed on Import**

An option set in Tools - Options by default trims leading and trailing white space from strings. This may be turned off if desired. The trim function trims white space so that strings like " a b cdef " (with spaces before the a and spaces after the f) are trimmed to "a b cdef". Internal whitespace is unaffected.

**Typical Drawing Formats**

Manifold can import from many different drawing formats. Some are highly specialized formats that provide much data organized in a complex way. Precise description of certain specialty formats is beyond the scope of this document. In such cases, please refer to the originating agency or company for detailed documentation on their format.

Interested in support for a new format not listed below? See the Contacting manifold.net topic for information on suggesting a new format.

Notes on a few of the most popular formats are set forth in the following topics:

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADF</td>
<td>Import a drawing from ESRI .adf ArcGrid format.</td>
</tr>
<tr>
<td>ADO .NET Data Sources</td>
<td>Create a drawing by importing data from a table using an ADO .NET, ODBC or OLE DB data source. Imported using the Data Source dialog.</td>
</tr>
<tr>
<td>BNA</td>
<td>Atlas .bna format.</td>
</tr>
<tr>
<td>CSV</td>
<td>Create a drawing by importing data from a geocoded table from a text file using comma-separated value format. Imported using the Data Source dialog.</td>
</tr>
<tr>
<td>DB</td>
<td>Create a drawing by importing data from a geocoded table from a Borland Paradox file. Imported using the Data Source dialog.</td>
</tr>
<tr>
<td>DBF</td>
<td>Create a drawing by importing data from a geocoded table from a dBase or FoxPro file. Imported using the Data Source dialog.</td>
</tr>
<tr>
<td>DGN</td>
<td>Bentley MicroStation / Intergraph .dgn.</td>
</tr>
<tr>
<td>DLG</td>
<td>USGS DLG optional format.</td>
</tr>
<tr>
<td>DWG</td>
<td>AutoCAD drawing format.</td>
</tr>
<tr>
<td>DXF</td>
<td>AutoCAD exchange format.</td>
</tr>
<tr>
<td>DSN</td>
<td>Create a drawing by importing data from a geocoded table using a DSN file data source. Imported using the Data Source dialog.</td>
</tr>
<tr>
<td>E00</td>
<td>ESRI ArcInfo export format. Manifold will automatically detect and correctly import drawings, images, surfaces and labels within the .e00.</td>
</tr>
<tr>
<td>GDF</td>
<td>Import TeleAtlas MultiNet in GDF format. Also used to</td>
</tr>
</tbody>
</table>
Import and Export

import European GDF format.

Geocoding Database
Import drawings of US streets on a county basis from the Manifold Geocoding Database provided on the Manifold downloads site. Requires installation of the Manifold Geocoding Tools extension.

Geodatabase
Import or link drawings from ESRI geodatabase files. Use Database Console to connect (via the Data Source dialog) to the .mdb file hosting the Personal Geodatabase and then Manifold's spatial DBMS capability will automatically identify the contents of that geodatabase. We can then import or link drawings as desired. See the spatial DBMS topic.

GML
Geography Markup Language. A spectacularly inefficient format based on XML that is being touted as a "standard" in some circles. Used by the UK's Ordnance Survey.

HTML
Create a drawing by importing data from a geocoded table from an HTML file. Imported using the Data Source dialog.

KML, KMZ
An XML-based format used by Google Earth and others. KMZ is a "zip" compressed version of KML.

LULC
USGS Land Use and Land Cover format.

Etak MapBase
Create a drawing by importing data from a geocoded table from a Microsoft Access file. Also used to import or link drawings from ESRI geodatabase files. Imported using the Data Source dialog.

MFD
Manifold System 4.50 format. Will import objects from an .mfd even if the associated .mdb database file is not present.

MIF
MapInfo .mid / .mif format. Will import objects from a .mif even if the associated .mid database file is not present.

MWS
Manifold System 4.5 .mws format.

NTAD
Bureau of Transportation Statistics National Atlas Transportation Database (NTAD) and North American Transportation Atlas Data (NORTAD) files.

NTF
UK Ordnance Survey .ntf files. Will automatically import Land-Line vector data sets as drawings and DTM grids as surfaces.

ODBC Data Sources
Create a drawing by importing data from a table using an ODBC data source. Imported using the Data Source dialog.

OLE DB Data Sources
Create a drawing by importing data from a table using an OLE DB data source. Imported using the Data Source dialog.

S-57
A simple importer for vector data in S-57 format (used for nautical charts). Manifold's importer attempts to bring in as much data as possible, leaving it up to the user to format and organize the data. Should greater interest appear in this format and enough user requests are received (see the Contacting manifold.net topic), the S-57 importer can always be tuned to provide a higher degree of organization or formatting in support of particular S-57 data sets.

SDTS
US Federal Spatial Data Transfer Standard. Can include drawings, surfaces, tables and comments. The correct component will automatically be created based on the
contents of the SDTS files.

**SHP**
ESRI ArcView .shp ("shapefile") format.

**TAB**
MapInfo "table" format consisting of multiple files (usually 5) referenced by a controlling .tab file. Can result in drawings, labels, images and surfaces.

**TAIF**
TeleAtlas TAIF format. Choose the TAIF subtype (Level 1, Level 2, Multinet or Streetnet) and the module desired. Map fields between tables and Import Z are enabled only for Multinet. Same function as SDTS. Will create multiple drawings and map.

**TIGER/Line**
US Bureau of the Census TIGER/Line formats.

**IDRISI VCT**
IDRISI .vct. Import a vector drawing from IDRISI 32 without any data attributes. Does not import from IDRISI 16 format.

**UDL**
Create a drawing by importing data from a geocoded table from a Microsoft Universal Data Link file. Imported using the Data Source dialog.

**VMAP**
NIMA (www.nima.mil) VMAP 0, VMAP 1 and VMAP 2 format, based on .vpf. Choose coverages to be imported. Import fields will import all data attributes associated with a coverage. Import Z causes an additional drawing to be created filled with points each of which has a Z coordinate. Will create multiple drawings, labels and maps.

**WKx**
Create a drawing by importing data from a geocoded table from a Lotus file. Imported using the Data Source dialog.

**XLS**
Create a drawing by importing data from a geocoded table from an Excel file. Imported using the Data Source dialog.

**DB2 Data Sources**
Connect to IBM DB2 equipped with the DB2 Spatial Extender using IBM's native spatial connection technology. A spatial DBMS connection accessed through the Data Source dialog. This option requires Enterprise Edition or above.

**Oracle Data Sources**
Connect to an Oracle data sources using the Oracle Call Interface (OCI), the native Oracle interface for exchanging data. A spatial DBMS connection accessed through the Data Source dialog. This option requires Enterprise Edition or above.

**PostgreSQL Data Sources**
Connect to PostgreSQL equipped with the PostGIS spatial extender using native PostgreSQL connection technology. Connections made using this data source will have any password required masked so that later usage will not inadvertently expose the password. A spatial DBMS connection accessed through the Data Source dialog. This option requires Enterprise Edition or above.

**SQL Server Data Sources**
Connect to Microsoft SQL Server 2008 spatial DBMS facilities using native SQL Server 2008 spatial connection technology. A spatial DBMS connection accessed through the Data Source dialog. This option requires Enterprise Edition or above.

**Important:** Using a vendor's native spatial DBMS connection technology (like those above) provides better performance and allows using features, such as built-in spatial DBMS facilities, not exposed through generic database interfaces such as ODBC, OLE DB and ADO .NET. For example, exchanging data with Oracle data sources via OCI automatically maps geometry columns.
Import and Export

into Oracle SDO_GEOMETRY data and will likewise map geometry columns into the native geometry types used by DB2, PostgreSQL and SQL Server 2008.

Note: because the Database Console provides a more convenient user interface for browsing databases, we will almost always use the Database Console to import or link components from a database instead of using the File - Import or File - Link menu commands.

Importing Drawings from Geocoded Tables

Drawings may also be created from geocoded tables by using File - Import - Drawing and selecting Data Sources in the Files of type box to launch the Data Source dialog. See the Linked Drawings from Geocoded Tables topic for a general introduction.

In the case of imports a drawing of points will be created from the geocoded table but it will not be linked to the table. Linked drawings are read-only because their content is controlled entirely by the external table but a drawing imported from a geocoded table will be read / write because it is an ordinary drawing like any other in a project. Unlike a linked drawing, however, any change to the originating geocoded table will not be reflected in the imported drawing.

Formats from which geocoded tables may be read to create drawings include:

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSV</td>
<td>Comma separated values ASCII files, including files using .txt or .csv or other extensions.</td>
</tr>
<tr>
<td>DB</td>
<td>Borland Paradox.</td>
</tr>
<tr>
<td>DBF</td>
<td>dBase and FoxPro.</td>
</tr>
<tr>
<td>DSN</td>
<td>ODBC file data source that describes an ODBC driver to use.</td>
</tr>
<tr>
<td>HTML</td>
<td>Tables in web pages.</td>
</tr>
<tr>
<td>UDL</td>
<td>Microsoft Universal Data Link file that describes an OLE DB connection.</td>
</tr>
<tr>
<td>WKx</td>
<td>Lotus tables.</td>
</tr>
<tr>
<td>XLS</td>
<td>Excel tables.</td>
</tr>
</tbody>
</table>

ADO.NET Data Sources

A generic ADO.NET connection allowing use of a .NET DLL assembly, provider class and connection string (parameters).

ADO.NET ODBC Data Sources

Connect using the standard Microsoft .NET ADO.NET provider for ODBC.

ADO.NET OLE DB Data Sources

Connect using the standard Microsoft .NET ADO.NET provider for OLE DB.

ADO.NET Oracle Data Sources

Connect using the standard Microsoft .NET ADO.NET provider for Oracle.

ADO.NET SQL Server Data Sources

Connect using the standard Microsoft .NET ADO.NET provider for SQL Server.

ODBC Data Sources

Connect to a data source using ODBC drivers installed on this system.

OLE DB Data Sources

Connect to an OLE DB data source using an OLE DB provider installed on this system.
Oracle Data Sources

Connect to an Oracle data sources using the Oracle Call Interface (OCI), the native Oracle interface for exchanging data. Using the native interface provides better performance and allows using features not exposed through generic database interfaces such as ODBC, OLE DB and ADO.NET. Exchanging data with Oracle data sources via OCI automatically maps geometry columns into Oracle SDO_GEOMETRY data.

Note that the table used to create the drawing must be a geocoded table, that is, each record must contain latitude and longitude values for that record's position expressed in decimal degrees. By default, drawings imported from geocoded tables consist of points, one point for each record, although if the table's content permits lines also may be optionally created. See the Linked Drawings topic for more information.

Importing Drawings from a Spatial DBMS

A more sophisticated way of storing drawings in databases is to store actual geometry data within a spatial DBMS. Manifold supports the major spatial DBMS players directly, including Oracle Spatial, IBM DB2 with Spatial Extender, Microsoft SQL Server 2008 spatial, SQL Server 2005 with the Manifold spatial extender, ESRI SDE geodatabases, ESRI Personal geodatabases and just about any other DBMS using spatial DBMS capabilities conferred by Manifold through generic spatial indices.

To import drawings from such databases we use the Database Console dialog to connect to the data source (using the Data Source dialog) and then import by highlighting the desired drawing and clicking the Import button in Database Console.

Note that connecting to some spatial DBMS products, such as using a native OCI connection to connect to Oracle Spatial, requires Enterprise Edition or above. See the Oracle Spatial Facilities topic for additional information and examples.

XML Files Created upon Export and Used on Import

Not all formats to which Manifold can export have the ability to correctly save projection information. As a safety measure to ensure that projection information is never lost, Manifold always writes an accompanying .xml file when exporting drawings, images, labels, maps or surface components to a file. The accompanying .xml file contains coordinate system information and some other metadata. The XML file is created for all formats, even those that correctly save coordinate system information.

When Manifold imports drawings, images, labels or surfaces, the system will check for an accompanying .xml file that might have been written by Manifold. If such an .xml file exists, Manifold will read it and use the information it contains to load a correct coordinate system.

See the XML Accessory File Format topic for details on the internal organization of this file.

Important Note when Using 64-bit Manifold Editions

Due to a lack of required Microsoft facilities in 64-bit Windows systems, Manifold in 64-bit mode cannot export, import, export or link to DB, HTML, MDB, XLS or WKx format files. This includes no access to the MDB parts of Manifold MFD and MapInfo TAB imports. The workaround for importing or exporting such files is to launch Manifold in 32-bit mode by using the Manifold System (32-bit) shortcut, perform the export from or import into a .map project file and then re-launch Manifold in 64-bit mode using the Manifold System (64-bit) shortcut.

Linking is more complex: the data must be in some format usable in 64-bit mode within 64-bit Windows systems, such as a SQL Server database. Alternatively, the data can be kept within a Manifold .map project file and linked using the Manifold ODBC driver.

Tech Tip

For advanced users: Some transform toolbar operators in drawings must assure that there are no errors in object metrics before running, so they run a normalization/error correction on all objects that are not already marked "clean" by Manifold. This can result in an unexpectedly slow operation on a newly imported drawing the first time such operators are run. Some drawings may be imported from sources that do not maintain clean object metrics. Expert users may want to run Normalize Topology or Normalize Metric on newly imported drawings just after the import to be sure that all objects are marked "clean" for future passes by any operators.
Import Drawing - ADF
Import ESRI .adf format files. ESRI .adf is sometimes called ArcGrid format.

.adf files can contain raster data, ESRI TIN data or vector (drawing) data. The Manifold .adf importer will automatically configure itself to import either drawings or surfaces as applicable regardless of whether the importer was launched via File - Import - Drawing or via File - Import - Surface. For information on importing .adf files that contain TINs or raster data, see the Import Surface - ADF topic. This topic discusses import of drawings.

Importing a Drawing

Importing an .adf file that contains vector data allows importing fields (data attributes) via the following dialog options:

- **Select All** - Check all fields. All data fields will be imported with this drawing.
- **Select None** - Uncheck all fields. No data fields will be imported. Tech tip: Use this to uncheck all fields before checking those desired.
- **Select Inverse** - Uncheck all previously checked fields and check all previously unchecked fields. This is a fast way to check only a few columns out of many: uncheck those desired and then push Select Inverse.

(Fields pane) Check the data attribute fields to be imported.

- **Compose areas** - Create areas (checked by default). Not enabled when the file contains no areas. May be unchecked if areas are not needed. Importing areas is a computationally intensive process so not importing them speeds up the import process.
- **Import ticks** - Import ESRI tick marks.

See Also

Import Surface - ADF
Import Drawing - ADO .NET / ODBC / OLE DB

Manifold can create a drawing directly from data stored in a geocoded table. This capability is directly analogous to the creation of a linked drawing from a geocoded table except that instead of the drawing being linked to the originating table it is created as an independent drawing within the Manifold project.

Such imports are accomplished by choosing File - Import - Drawing and then choosing Data Sources () in the Files of type box to launch the Data Source dialog. The desired data source can then be configured and used in the Data Source dialog.

Drawings can be imported from geocoded tables using any of the data access methods, such as connection to an ODBC or OLE DB data source, provided by Manifold.

Dialogs used are the same as for creating a linked drawing from a geocoded table, except that linked drawings always invoke the Data Source dialog and the result of an import is an imported drawing rather than a linked drawing.

See Also

The Data Source Dialog
Importing and Linking Tables
Linked Drawings
Create a Linked Drawing from a Geocoded Table
Import Drawing - BNA
Import drawings in Atlas .bna format. The only option with this importer is which data fields should be imported from the Atlas drawing.

Select All - Check all fields. All data fields will be imported with this drawing.

Select None - Uncheck all fields. No data fields will be imported. Tech tip: Use this to uncheck all fields before checking those desired.

Select Inverse - Uncheck all previously checked fields and check all previously unchecked fields. This is a fast way to check only a few columns out of many: uncheck those desired and then push Select Inverse.

Fields pane Check the data attribute fields to be imported.
Import and Export

Import Drawing - CSV

Manifold can create a drawing directly from data stored in a geocoded table. This capability is directly analogous to the creation of a linked drawing from a geocoded table except that instead of the drawing being linked to the originating table it is created as an independent drawing within the Manifold project.

Such imports are accomplished by choosing File - Import - Drawing and then choosing Data Sources ( ) in the Files of type box to launch the Data Source dialog. The desired data source can then be configured and used in the Data Source dialog. Note that the CSV import choice includes files with any extension. For example, a .txt file that contains a table in comma-separated value format can be used with this importer.

Dialogs used are the same as for creating a linked drawing from a geocoded table, except that linked drawings always invoke the Data Source dialog and the result of an import is an imported drawing rather than a linked drawing.

See Also

The Data Source Dialog
Importing and Linking Tables
Linked Drawings
Create a Linked Drawing from a Geocoded Table
Import Drawing - DB

Manifold can create a drawing directly from data stored in a geocoded table. This capability is directly analogous to the creation of a linked drawing from a geocoded table except that instead of the drawing being linked to the originating table it is created as an independent drawing within the Manifold project.

Such imports are accomplished by choosing File - Import - Drawing and then choosing Data Sources (\) in the Files of type box to launch the Data Source dialog. The desired data source can then be configured and used in the Data Source dialog.

Dialogs used are the same as for creating a linked drawing from a geocoded table, except that linked drawings always invoke the Data Source dialog and the result of an import is an imported drawing rather than a linked drawing.

Important Note when Using 64-bit Manifold Editions

Due to a lack of required Microsoft facilities in 64-bit Windows systems, Manifold in 64-bit mode cannot export, import, export or link to DB, HTML, MDB, XLS or WKx format files. This includes no access to the MDB parts of Manifold MFD and MapInfo TAB imports. The workaround for importing or exporting such files is to launch Manifold in 32-bit mode by using the Manifold System (32-bit) shortcut, perform the export from or import into a .map project file and then re-launch Manifold in 64-bit mode using the Manifold System (64-bit) shortcut. Linking is more complex: the data must be in some format usable in 64-bit mode within 64-bit Windows systems, such as a SQL Server database. Alternatively, the data can be kept within a Manifold .map project file and linked using the Manifold ODBC driver.

See Also

The Data Source Dialog
Importing and Linking Tables
Linked Drawings
Create a Linked Drawing from a Geocoded Table
**Import Drawing - DBF**

Manifold can create a drawing directly from data stored in a geocoded table. This capability is directly analogous to the creation of a linked drawing from a geocoded table except that instead of the drawing being linked to the originating table it is created as an independent drawing within the Manifold project.

Such imports are accomplished by choosing **File - Import - Drawing** and then choosing **Data Sources (]** in the **Files of type** box to launch the Data Source dialog. The desired data source can then be configured and used in the Data Source dialog.

Drawings can be imported from geocoded tables using any of the data access methods, such as from a dBase or FoxPro **DBF** file, provided by Manifold. Dialogs used are the same as for creating a linked drawing from a geocoded table, except that linked drawings always invoke the Data Source dialog and the result of an import is an imported drawing rather than a linked drawing.

**See Also**

- The Data Source Dialog
- Importing and Linking Tables
- Linked Drawings
- Create a Linked Drawing from a Geocoded Table
Import Drawing - DGN

Bentley MicroStation Design File format (DGN) is also known as Intergraph DGN format due to Intergraph's sales of Bentley GIS products. Current editions of Manifold import Version 7 of DGN. Future editions will likely add support for Version 8.

Importing a drawing from DGN format allows two options:

**Threshold**  
Threshold expressed in whatever units are used within the .dgn. Within this threshold straight lines are used to approximate curved elements such as splines or circles.

**Import Z**  
Create an additional drawing with points that provide a Z coordinate.

Several formats in common use allow objects such as lines to be defined not just by X,Y coordinates but also by Z coordinates at each coordinate location that defines the line. Checking the Import Z box causes an additional drawing to be created filled with points, each of which has a Z coordinate taken from the accessory Z values in the coordinate sequences that define lines or other objects. This drawing can then be used to create a surface. Note: if there are several sets (layers) of objects in a original .dgn, all of the Z points will be created in one Z drawing. Use spatial overlays to select/filter only those Z points desired for a given set of objects.

Importing a DGN into Manifold will import splines as curved shapes.

Importing a DGN file with external references will cite the names of any reference files in the History pane.

**Note:** DGN files are not really geographic files but utilize an unusual internal coordinate system unique to MicroStation. Making sense of such data requires formidably expert skills in MicroStation. MicroStation experts advise a simpler path, writing MicroStation data to AutoCAD DWG, making sure to write the data out in master units and thus forcing MicroStation to write out in sensible XYZ coordinates instead of the bizarre internal coordinates used by MicroStation. One can then import the AutoCAD DWG file using the usual procedures when importing projected information from non-geographic formats. See the Import a Projected Shapefile example for an example of such procedures.

**Broken DGN Files**

DGN may be imported by Manifold in cases where the .dgn is not “broken” (to use Bentley’s terminology). In cases of “broken” .dgn files, importing a .dgn may show elements in the Manifold drawing that were recently deleted in MicroStation before saving the .dgn.

To avoid this situation, try the following:

- In MicroStation, choose **File - Compress Design** to compress the .dgn before it is imported into Manifold.
- If there are still problems, use the **EdG** utility provided with MicroStation to repair the broken .dgn.

**Notes**

DGN is thinly documented and can include format variations that are not openly documented. Although Manifold’s .dgn importer has proven to work very well across a wide range of .dgn files it is always possible that a .dgn may be encountered in an undocumented format variation that cannot be imported into Manifold.

If such files are encountered, the best approach is to track down the person or organization that created them and to convince them to switch to Manifold for all of their GIS needs. This is especially a good thing to do if the organization is a large one and the result is acquisition of thousands of Manifold licenses.

The second-best approach is to ask the issuing person or organization to write out the data in some other GIS format that can be read by Manifold, such as AutoCAD DWG, an ESRI format such as shapefiles or some neutral format such as SDTS.
Import and Export

Import Drawing - DLG

A U.S. Geological Survey (USGS) format, Digital Line Graph (DLG) format is now being replaced by SDTS format. However, many drawings are still available in DLG format. Manifold's DLG importer imports so-called "DLG Optional" format, the DLG format version which replaced the original DLG format used many years ago. Virtually all "DLG" files available use the more modern "optional" format; however, on rare occasion one may encounter very old "dlg" files in the older format. DLG files should have extensions such as .dlg, .do, or .opt.

Select All - Check all fields. All data fields will be imported with this drawing.

Select None - Uncheck all fields. No data fields will be imported. Tech tip: Use this to uncheck all fields before checking those desired.

Select Inverse - Uncheck all previously checked fields and check all previously unchecked fields. This is a fast way to check only a few columns out of many: uncheck those desired and then push Select Inverse.

Fields pane Check the data attribute fields to be imported.

Compose areas Check to create area objects. Not checking this box will create only lines and points.

Tech Tips

In 1993 USGS published a series of 14 CDs providing 1:100K-scale DLGs covering the entire US. These CDs contain DLGs in zipped format. The DLG files must be first unzipped. The result is files that are named using numbers without a three letter extension, for example SF4RDF04. These must be renamed to add a .opt extension to each file, for example SF4RDF04.opt.

Trouble importing DLGs? Check the obvious:

- Are they in DLG optional format? "Original" format DLG files are extremely rare but may still occur.
- Have you unzipped or otherwise decompressed the files if they were provided in a compressed way?
- Are they really DLG or are they SDTS format? Many files provided by USGS as DLGs are now provided in SDTS format, but the directory structures or other descriptions refer to them as "DLGs" because they were originally provided in DLG format.
- If you have downloaded files via the Internet, try downloading again in case the files were damaged during download.

Manifold will try to import even malformed DLGs. For example, Manifold will automatically handle inconsistent line ends found in malformed DLGs.
Import Drawing - DSN

Manifold can create a drawing directly from data stored in a geocoded table. This capability is directly analogous to the creation of a linked drawing from a geocoded table except that instead of the drawing being linked to the originating table it is created as an independent drawing within the Manifold project.

A Microsoft file DSN is a way of capturing an ODBC connection within a simple connection that may be easily "opened" without the need to deal with complex ODBC dialogs (except once when the DSN is first set up).

Such imports are accomplished by choosing File - Import - Drawing and then choosing Data Sources () in the Files of type box to launch the Data Source dialog. The desired data source can then be configured and used in the Data Source dialog.

Dialogs used are the same as for creating a linked drawing from a geocoded table, except that linked drawings always invoke the Data Source dialog and the result of an import is an imported drawing rather than a linked drawing.

See Also

The Data Source Dialog
Importing and Linking Tables
Linked Drawings
Create a Linked Drawing from a Geocoded Table
Import and Export

Import Drawing - DWG

AutoCAD .dwg ("drawing") format is a CAD format that is occasionally encountered in GIS, although AutoCAD .dxf ("export") format is more frequently encountered. The format is a first rate choice for CAD work but must be imported with care in GIS work since it saves neither database information nor projection settings. Not surprisingly, .dwg format is used mostly to save CAD work such as diagrams of facilities and other CAD drawings created without any sense of projection or geographic location.

However, one also encounters drawings saved in .dwg where the CAD system was used in an attempt to render cartographic features intended to have meaning within a particular projection or tied to specific geographic locations. The popularity of AutoCAD assures that there will be many drawings saved in .dxf that one may wish to use in Manifold.

Manifold's .dwg importer can import versions 13, 14 and 15 of .dwg (version 15 is produced by AutoCAD 2000).

- **Select All** - Check all the boxes in the layers pane.
- **Select None** - Uncheck all the boxes in the layers pane.
- **Select Inverse** - Check all unchecked boxes and uncheck all checked boxes in the layers pane.

**Layers**

- **Check layers to be imported.**

**Threshold**

Threshold expressed in whatever units are used within the .dwg. Within this threshold straight lines are used to approximate curved elements such as splines or circles. In general, the threshold parameter specifies the maximum error that can be made when approximating curved elements with straight lines in units used within the file.

**Import lines with widths as areas**

Import lines that have non-zero widths as area objects, using the width of the area to represent the width of the line.

**Import Z**

Create an additional drawing with points that provide a Z coordinate.

**Strip format tags**

Remove formatting information from label texts. On by default.

Several formats in common use allow objects such as lines to be defined not just by X,Y coordinates but also by Z coordinates at each coordinate location that defines the line. Checking the **Import Z** box causes an additional drawing to be created filled with points, each of which has a Z coordinate taken from the accessory Z values in the coordinate sequences that define lines or other objects. This drawing can then be used to create a surface. Note: if there are several sets (layers) of objects in the original .dwg, all of the Z points will be created in one Z drawing. Use spatial overlays to select/filter only those Z points desired for a given set of objects.

.dwg drawings will often include many layers. Each layer checked for import will be imported into Manifold as a separate drawing so that they can appear within a map as layers. Manifold will attempt to name the drawing using the layer name within the .dwg file.

To deal with projection issues when importing .dwg, see the Import Drawing - DXF topic, which provides a discussion using import from .dxf files.

Importing a DWG file with external references will cite the names of any reference files in the History pane.
AutoCAD .dxf format is a CAD format that is frequently encountered in GIS. The format is a first rate choice for CAD work but must be imported with care in GIS work since it saves neither database information nor projection settings. Not surprisingly, .dxf format is used mostly to save CAD work such as diagrams of facilities and other CAD drawings created without any sense of projection or geographic location.

However, one also encounters drawings saved in .dxf where the CAD system was used in an attempt to render cartographic features intended to have meaning within a particular projection or tied to specific geographic locations. The popularity of AutoCAD assures that there will be many drawings saved in .dxf that one may wish to use in Manifold.

Manifold’s .dxf importer can import versions 13, 14 and 15 of .dxf (version 15 is produced by AutoCAD 2000).

- **Select All** - Check all the boxes in the layers pane.
- **Select None** - Uncheck all the boxes in the layers pane.
- **Select Inverse** - Check all unchecked boxes and uncheck all checked boxes in the layers pane.

**Layers** Check layers to be imported.

**Threshold** Threshold expressed in whatever units are used within the .dxf. Within this threshold straight lines are used to approximate curved elements such as splines or circles. In general, the threshold parameter specifies the maximum error that can be made when approximating curved elements with straight lines in units used within the file.

**Import lines with widths as areas** Import lines that have non-zero widths as area objects, using the width of the area to represent the width of the line.

**Import Z** Create an additional drawing with points that provide a Z coordinate.

**Strip format tags** Remove formatting information from label texts. On by default.

.dxf drawings will often include layers. Each layer check will be imported into Manifold as a separate drawing so that they can appear within a map as layers. A map component containing all the layers will also be created. Manifold will attempt to name the drawing using a combination of the file name and the layer name within the .dxf file.

Several formats in common use allow objects such as lines to be defined not just by X,Y coordinates but also by Z coordinates at each coordinate location that defines the line. Checking the **Import Z** box causes an additional drawing to be created filled with points, each of which has a Z coordinate taken from the accessory Z values in the coordinate sequences that define lines or other objects. This drawing can then be used to create a surface. Use spatial overlays to select/filter only those Z points desired for a given set of objects.

Importing a DGN, DWG or DXF file with external references will cite the names of any reference files in the History pane.

**Projections and DXF**

All .dxf drawings imported into Manifold will require some manual setting of coordinate properties using the Edit - Assign Projection dialog. Each drawing created from the .dxf (in the case of multiple drawings created when there are multiple layers in the .dxf) will have to have its coordinates properties correctly set. This is tedious.

.dxf drawings will initially be imported using the default Orthographic projection centered at 0 latitude and 0 longitude. To place the drawing into correct geographic context we must provide projection parameters to the
Edit - Assign Projection dialog. To do this we must find out what projection (if any) was used to create the .dxf file.

There are essentially two main classes of .dxf drawings we will import into Manifold:

- CAD drawings created in an Euclidean CAD space using meters or feet or other linear units. These may be various blueprints or other diagrams showing facilities, layout of land parcels and so on. What they have in common is that there is no pretext to show large geographic areas that might involve projections.
- Geographic drawings rendered in .dxf. These may have been imported from some other GIS format. They may or may not show geographic features using a projection.

To be used in a geographic context with other maps and drawings, CAD drawings must be georegistered using features they have in common with an already-georegistered drawing or image. For example, if we have a .dxf drawing of a warehouse facility we will need to find a drawing that shows features that are also found in the drawing of the warehouse facility that may be used as control points. We can then use these common features (such as the corners of the parcel or the location of roads shown both in the CAD drawing and in our georeferenced drawing) as control points.

The process for CAD drawings is to import them using defaults. We can then georegister them. If we don't care about the geographic context of the drawings, we can always use Manifold as a CAD editor by simply leaving the CAD drawings in their default imported state and editing them as we might within AutoCAD.

To use geographic drawings saved in .dxf formats within Manifold, we have basically three situations to deal with:

- Drawings created in Latitude / Longitude unprojected form using degrees as units of measure.
- Drawings created in some projection using feet, meters or some other linear unit of measure.
- Drawings where the projection used is unknown.

For the third case, we have no choice but to attempt to use georegistration to place them in the correct geographic context. For the other two cases, we can proceed as set forth in the following two examples.

Example: Importing an Unprojected DXF

Suppose we know the .dxf being imported was created in the Latitude / Longitude projection using degrees as units of measurements. This is a fairly rare event in the AutoCAD world since most CAD projects, even geographic ones, are created using meters or feet or some other linear unit of measure.

In our example we import a .dxf drawing of the United States. Note that .dxf cannot save areas, so .dxf drawings consist of lines. On import, we can open the drawing and see that it appears like a normal, unprojected view of the United States even though the status bar tells us it is in Orthographic projection.

| Orthographic | 0°00'03"W, 0°00'01"N |

On moving the mouse cursor over the drawing, the status bar will show that the coordinate system in use is wrong.
We can open the Edit - Assign Projection dialog to see what projection settings Manifold is using to interpret the imported drawing. We will use this dialog to change the settings.
All we need do is select Latitude / Longitude from the projections hierarchy and then press OK. The drawing will now be correctly interpreted.

Example: Importing a Projected DXF

The usual situation with .dxf files showing geographic entities is that they are created in some projection, usually a bizarre and painful projection such as UTM. For this example, we assume we have imported a .dxf drawing that we know represents a map in Lambert Conformal Conic projection. Suppose we further know that the projection was centered on latitude 40, longitude -100 using a first standard parallel of 33 and a second standard parallel of 45.

We import the drawing using defaults.
When we click open the drawing we see that it appears to show the United States in typical **Lambert Conformal Conic** projection even though the status bar says the projection in use is **Orthographic**.

![Orthographic Projection](image)

When we pass the mouse cursor over the drawing window we see that the status bar reports nonsensical latitude and longitude values for the central US.

As in the previous example, we can open the **Edit - Assign Projection** dialog to see what projection settings Manifold is using to interpret the imported drawing. They are the same as the previous example since .dxf drawings are always imported by default into **Orthographic** projection.
We choose \textit{Lambert Conformal Conic} as the projection from the hierarchy of projections available. We then double-click into each of the values to change them to the values we know were used in the projection of the drawing. We enter 40 for the \textit{Center Latitude} and -100 for the \textit{Center Longitude}.

"Parallel" is a synonym for "latitude." We therefore enter 33 and 45 as the \textit{1st Standard Latitude} and \textit{2nd Standard Latitude} respectively. The illustration above shows the dialog as we enter 33 for the \textit{1st Standard Latitude}. Press \textit{OK} to apply the values. The drawing will now be correctly interpreted.

\textbf{Comments}

Where did we get the magic values that were entered into the dialog to make sense of the projected .dxf? Because .dxf format does not save projection information we must get these magic values from the creator of the .dxf file. Whenever downloading or otherwise procuring .dxf files that may be projected, it is absolutely critical to stay alert for any accompanying documentation.

Suppose we don't have the accompanying documentation? Suppose we know that the .dxf represents a projected drawing but we have no idea what projection was actually used? In that case, we pause to curse the inconsiderate nature of people who use .dxf formats to save projected drawings without keeping their users informed and then we move on to using georegistration to register the drawing within a geographic context. We do this by matching up features it has in common with a "known good" drawing using control points.

Georegistration is a real hassle. Adding control points is tedious. A more serious problem is that inevitably some slight precision is lost when georegistering as compared to using the original data using precise projection parameters as originally intended, even though any loss of precision is normally far less than the intrinsic accuracy of the drawing. For these reasons it is wise to spare no effort in tracking down a description of the projection used to create a projected .dxf drawing.
Import Drawing - E00

ESRI .e00 is a complex format used with ArcINFO. It is capable of storing different types of data, including images and surfaces ("grids"), but is most often used for drawings. .e00 format occurs in single-precision or double-precision forms and both compressed (binary) and uncompressed (ASCII) types.

The Manifold .e00 importer can automatically determine which form of .e00 is being opened and extract the data it contains. .e00 import will also automatically import any auxiliary attribute tables and label sets.

- **Select All** - Check all fields. All data fields will be imported with this drawing.
- **Select None** - Uncheck all fields. No data fields will be imported. Tech tip: Use this to uncheck all fields before checking those desired.
- **Select Inverse** - Uncheck all previously checked fields and check all previously unchecked fields. This is a fast way to check only a few columns out of many: uncheck those desired and then push **Select Inverse**.

**Fields pane** Check the data attribute fields to be imported.

**Compose areas** Create areas (checked by default). Not enabled when the file contains no areas. May be unchecked if areas are not needed. Importing areas is a computationally intensive process so not importing areas will speed up the import process.

**Import Ticks** Create an additional drawing with tick points

**Comments**

.e00 files are sometimes multiple files, where a single data set is contained as a series of files named filename.e00, filename.e01, filename.e02 and so on. When downloading ".e00" files be on the lookout for such additional files and download them all.

.e00 is an undocumented format that apparently has been changed by ESRI over time. Although Manifold's .e00 import and export capabilities will obviously attempt to follow any changes introduced by ESRI, there is no guarantee that what works today with current versions of ESRI products will also work with future ESRI products that did not exist at the time this Manifold release was created. For best compatibility with the widest range of ESRI products that can read or write .e00, make sure to upgrade to the latest Manifold release and service pack available.
Import Drawing - GDF

GDF is a format used for TeleAtlas MultiNet files. These are sometimes casually referred to as TeleAtlas Import Format (TAIF) files, although strictly speaking TAIF is a different format normally used for TeleAtlas StreetNet files.

GDF is also a generic European standard used for GIS data sets. Manifold's GDF importer has been verified with MultiNet drawings and will be extended to import all GDF files using the standard European format.

Select All - Check all modules. All modules will be imported with this drawing.

Select None - Uncheck all modules. No data modules will be imported. Tech tip: Use this to uncheck all modules before checking those desired.

Select Inverse - Uncheck all previously checked modules and check all previously unchecked modules. This is a fast way to check only a few modules out of many: uncheck those desired and then push Select Inverse.

Modules pane

Map fields between tables

Import Z

Several formats in common use allow objects such as lines to be defined not just by X,Y coordinates but also by Z coordinates at each coordinate location that defines the line. Checking the Import Z box causes an additional drawing to be created filled with points, each of which has a Z coordinate taken from the accessory Z values in the coordinate sequences that define lines or other objects. This drawing can then be used to create a surface. Note: if there are several sets (layers) of objects in the original file, all of the Z points will be created in one Z drawing. Use spatial overlays to select/filter only those Z points desired for a given set of objects.

See Also

Import Drawing - TAIF for import of files in TeleAtlas TAIF format.
Import and Export

Import Drawing - Geocoding Database

Drawings may be imported from the Manifold Geocoding Database that is provided on the Manifold downloads site. The geocoding database provides information on streets and addresses in the United States that is organized on a county basis, but it also includes supplementary cartographic information. Although the principal use of the geocoding database is to provide address range data for street address geocoding the data may also be used to create drawings that show streets and hydrological features within the United States.

Note that the Manifold Geocoding Database contains data only for the United States; therefore, drawings imported from the geocoding database can only be imported for counties within the United States.

Required Geocoding Tools and Geocoding Database

Like the rest of Manifold’s street address geocoding capability, import from the geocoding database becomes operational only when the optional Manifold Geocoding Tools package is installed. If you do not have the optional Geocoding Tools package, you will not be able to take advantage of the import capability described in this topic.

Note that Manifold Universal Edition automatically installs the Geocoding Tools package. If you are using Universal Edition, you do not need to perform an extra Geocoding Tools installation as it has already been installed.

To install the Manifold Geocoding Database to hard disk, see the instructions in the Geocoding Data Sources topic.

Importing a Drawing

Drawings are normally imported from the geocoding database after it has been installed to hard disk. See the Geocoding Data Sources topic for instructions on installing the Manifold Geocoding Database onto your system’s hard disk.

To import a drawing from the Manifold Geocoding Database:

1. Use File - Import - Drawing. In the Import Drawing dialog choose Geocoding Database Files in the Files of type box.
2. In the Import Drawing dialog browse over to the installation folder for the geocoding database (by default, C:\Program Files\Manifold System\GCDB) and double-click the states.dat file.
3. In the Import Geocoding Database dialog enter a list of one or more counties to be imported in the Counties box.
4. Check fields (columns) to be imported for each object. It is strongly recommended to import at least the CFCC field so that different types of objects can be thematically formatted and separated if need be. Press OK.

A drawing will be imported for each county specified. If the CFCC field has been selected (it is selected by default) a thematic format will be applied that colors hydrology features in blue, railroads in black and other features in neutral brown colors. Railroad line size will be thematically formatted at a larger size than other lines.

If we would like to combine objects from multiple county drawings into a single drawing we can use Copy and Paste to copy objects from one drawing and paste them into another drawing.

If we would like to separate features into different drawings, we can easily do so using the selection toolbar to select different ranges of CFCC codes, cutting the selected features out of the drawing with a CTRL-X and then pasting them as new drawings in the Project pane.

For an example of drawing import from a geocoding database and separation of features into different drawings, see the Import a Drawing from the Geocoding Database example topic.

Controls

Counties One or more counties to import. A county is specified as a county name with a two letter state abbreviation in the format Countname, StateAbbreviation. Place a
semicolon between counties if more than one county is specified. For example, if we wanted to import streets for Alameda county in California as well as streets for Contra Costa county in California we would enter Alameda, CA; Contra Costa, CA

Select All - Check all fields. All fields will be imported with this drawing.

Select None - Uncheck all fields. No data fields will be imported. Tech tip: Use this to uncheck all fields before checking those desired.

Select Inverse - Uncheck all previously checked fields and check all previously unchecked fields. This is a fast way to check only a few fields out of many: uncheck those desired and then push Select Inverse.

Fields pane  Check the fields to be imported.

Import formatting Check (the default) to import default formatting used for geocoding database objects, such as blue color for hydrological features.

Fields

Field names and contents are derived from the original data source, the US Bureau of the Census TIGER/Line database. See the Census Bureau documentation on TIGER/Line attributes for details.

Street  Street name.

CFCC  Census Feature Class Code, a code that classifies objects into different types. CFCC codes that begin with H, for example, refer to hydrological (water) features.

FRADDL  Beginning (From) address number on the left side of the street.

TOADDL  Ending (To) address number on the left side of the street.

FRADDR  Beginning (From) address number on the right side of the street.

TOADDR  Ending (To) address number on the right side of the street.

ZIPL  US Postal Service ZIP code on the left side of the street.

ZIPR  US Postal Service ZIP code on the right side of the street.

CITYL  City name on the left side of the street.

CITYR  City name on the right side of the street.

STATEL  State name on the left side of the street.

STATER  State name on the right side of the street.

COUNTYL  County name on the left side of the street.

COUNTYR  County name on the right side of the street.

Most fields come in Left (L) and Right (R) versions, for example ZIPL and ZIPR. These specify the values for a particular street on the left side of the street or on the right side of the street, left and right being understood given
the orientation of that particular street segment as it is internally defined from the first coordinate that defines the street to the last coordinate.

It may seem odd to think of a street that has different ZIP codes on opposite sides of the same street, or different city, county or state names on opposite sides of the same street but that situation does occur in streets that mark the boundary between different jurisdictions.

**CFCC Codes**

The following codes are used for **CFCC** classification by the US Bureau of the Census:

<table>
<thead>
<tr>
<th>CFCC</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A00</td>
<td>Road, major and minor categories unknown</td>
</tr>
<tr>
<td>A01</td>
<td>Road, unseparated</td>
</tr>
<tr>
<td>A02</td>
<td>Road, unseparated, in tunnel</td>
</tr>
<tr>
<td>A03</td>
<td>Road, unseparated, underpassing</td>
</tr>
<tr>
<td>A04</td>
<td>Road, unseparated, with rail line in center</td>
</tr>
<tr>
<td>A05</td>
<td>Road, separated</td>
</tr>
<tr>
<td>A06</td>
<td>Road, separated, in tunnel</td>
</tr>
<tr>
<td>A07</td>
<td>Road, separated, underpassing</td>
</tr>
<tr>
<td>A08</td>
<td>Road, separated, with rail line in center</td>
</tr>
<tr>
<td>A10</td>
<td>Primary road with limited access or interstate highway, major category</td>
</tr>
<tr>
<td>A11</td>
<td>Primary road with limited access or interstate highway, unseparated</td>
</tr>
<tr>
<td>A12</td>
<td>Primary road with limited access or interstate highway, unseparated, in tunnel</td>
</tr>
<tr>
<td>A13</td>
<td>Primary road with limited access or interstate highway, unseparated, underpassing</td>
</tr>
<tr>
<td>A14</td>
<td>Primary road with limited access or interstate highway, unseparated, with rail line in center</td>
</tr>
<tr>
<td>A15</td>
<td>Primary road with limited access or interstate highway, separated</td>
</tr>
<tr>
<td>A16</td>
<td>Primary road with limited access or interstate highway, separated, in tunnel</td>
</tr>
<tr>
<td>A17</td>
<td>Primary road with limited access or interstate highway, separated, underpassing</td>
</tr>
<tr>
<td>A18</td>
<td>Primary road with limited access or interstate highway, separated, with rail line in center</td>
</tr>
<tr>
<td>A20</td>
<td>Primary road without limited access, U.S. and State highway, major category</td>
</tr>
<tr>
<td>A21</td>
<td>Primary road without limited access, U.S. and State highways, unseparated</td>
</tr>
<tr>
<td>A22</td>
<td>Primary road without limited access, U.S. and State highways, unseparated, in tunnel</td>
</tr>
<tr>
<td>A23</td>
<td>Primary road without limited access, U.S. and State highways, unseparated, underpassing</td>
</tr>
</tbody>
</table>
A24  Primary road without limited access, U.S. and State highways, unseparated, with rail line in center
A25  Primary road without limited access, U.S. and State highways, separated
A26  Primary road without limited access, U.S. and State highways, separated, in tunnel
A27  Primary road without limited access, U.S. and State highways, separated, underpassing
A28  Primary road without limited access, U.S. and State highways, separated, with rail line in center
A30  Secondary and connecting road, State and county highways, major category
A31  Secondary and connecting road, State and county highways, unseparated
A32  Secondary and connecting road, State and county highways, unseparated, in tunnel
A33  Secondary and connecting road, State and county highways, unseparated, underpassing
A34  Secondary and connecting road, State and county highways, unseparated, with rail line in center
A35  Secondary and connecting road, State and county highways, separated
A36  Secondary and connecting road, State and county highways, separated, in tunnel
A37  Secondary and connecting road, State and county highways, separated, underpassing
A38  Secondary and connecting road, State and county highway, separated, with rail line in center
A40  Local, neighborhood, and rural road, city street, major category
A41  Local, neighborhood, and rural road, city street, unseparated
A42  Local, neighborhood, and rural road, city street, unseparated, in tunnel
A43  Local, neighborhood, and rural road, city street, unseparated, underpassing
A44  Local, neighborhood, and rural road, city street, unseparated, with rail line in center
A45  Local, neighborhood, and rural road, city street, separated
A46  Local, neighborhood, and rural road, city street, separated, in tunnel
A47  Local, neighborhood, and rural road, city street, separated, underpassing
A48  Local, neighborhood, and rural road, city street, separated, with rail line in center
A50  Vehicular trail, road passable only by four-wheel drive (4WD) vehicle, major category
A51  Vehicular trail, road passable only by 4WD vehicle, unseparated
A52  Vehicular trail, road passable only by 4WD vehicle, unseparated, in
<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A53</td>
<td>Vehicular trail, road passable only by 4WD vehicle, unseparated, underpassing</td>
</tr>
<tr>
<td>A60</td>
<td>Special road feature, major category used when the minor category could not be determined</td>
</tr>
<tr>
<td>A61</td>
<td>Cul-de-sac, the closed end of a road that forms a loop or turn around</td>
</tr>
<tr>
<td>A62</td>
<td>Traffic circle, the portion of a road or intersection of roads that form a roundabout</td>
</tr>
<tr>
<td>A63</td>
<td>Access ramp, the portion of a road that forms a cloverleaf or limited access interchange</td>
</tr>
<tr>
<td>A64</td>
<td>Service drive, road that provides access to businesses, facilities, and rest areas along limited-access highway</td>
</tr>
<tr>
<td>A65</td>
<td>Ferry crossing, the representation of a route over water that connects roads on opposite shores</td>
</tr>
<tr>
<td>A66</td>
<td>Ferry crossing, Passenger, Year Round</td>
</tr>
<tr>
<td>A68</td>
<td>Ferry Crossing, Vehicular, Seasonal</td>
</tr>
<tr>
<td>A69</td>
<td>Ferry Crossing, Vehicular, Year-Round</td>
</tr>
<tr>
<td>A70</td>
<td>Other thoroughfare, major category used when the minor category could not be determined</td>
</tr>
<tr>
<td>A71</td>
<td>Walkway, nearly level road for pedestrians, usually unnamed</td>
</tr>
<tr>
<td>A72</td>
<td>Stairway, stepped road for pedestrians, usually unnamed</td>
</tr>
<tr>
<td>A73</td>
<td>Alley, road for service vehicles, usually unnamed, located at the rear of buildings and property</td>
</tr>
<tr>
<td>A74</td>
<td>Driveway or service road, usually privately owned and unnamed, used as access to residences, etc., or as access to logging areas, etc.</td>
</tr>
<tr>
<td>A75</td>
<td>Road, Parking Area</td>
</tr>
<tr>
<td>B00</td>
<td>Railroad, major and minor categories unknown</td>
</tr>
<tr>
<td>B01</td>
<td>Railroad track, not in tunnel or underpassing</td>
</tr>
<tr>
<td>B02</td>
<td>Railroad track, in tunnel</td>
</tr>
<tr>
<td>B03</td>
<td>Railroad track, underpassing</td>
</tr>
<tr>
<td>B10</td>
<td>Railroad main track, major category</td>
</tr>
<tr>
<td>B11</td>
<td>Railroad main track, not in tunnel or underpassing</td>
</tr>
<tr>
<td>B12</td>
<td>Railroad main track, in tunnel</td>
</tr>
<tr>
<td>B13</td>
<td>Railroad main track, underpassing</td>
</tr>
<tr>
<td>B20</td>
<td>Railroad spur track, major category</td>
</tr>
<tr>
<td>B21</td>
<td>Railroad spur track, not in tunnel or underpassing</td>
</tr>
<tr>
<td>B22</td>
<td>Railroad spur track, in tunnel</td>
</tr>
<tr>
<td>B23</td>
<td>Railroad spur track, underpassing</td>
</tr>
<tr>
<td>B30</td>
<td>Railroad yard track, major category</td>
</tr>
<tr>
<td>B31</td>
<td>Railroad yard track, not in tunnel or underpassing</td>
</tr>
</tbody>
</table>
B32 Railroad yard track, in tunnel
B33 Railroad yard track, underpassing
B40 Railroad ferry crossing, route over water used by ships carrying train cars to connecting railroads on opposite shores, major category
B42 Subway or Metroline
B50 Other rail line; major category used alone when the minor category could not be determined
B51 Carline, a track for street cars, trolleys, and other mass transit rail systems
B52 Cog railroad, incline railway, or logging tram
C00 Miscellaneous ground transportation, not road or railroad; major and minor categories unknown
C10 Pipeline, major category used alone
C20 Power transmission line, major category used alone
C30 Other ground transportation that is not a pipeline or a power transmission line, major category
C31 Aerial tramway, monorail, or ski lift
D00 Landmark, major and minor categories unknown
D01 TANA/GDT code for a building that is not classified
D10 Military installation or reservation; major category used alone
D20 Multihousehold or transient quarters; major category used alone when minor category not determined
D21 Apartment building or complex
D22 Rooming or boarding house
D23 Trailer court or mobile home park
D24 Marina
D25 Crew-of-vessel area
D26 Housing facility for workers
D27 Hotel, motel, resort, spa, hostel, YMCA, or YWCA
D28 Campground
D29 Shelter or mission
D30 Custodial facility; major category used alone when the minor category could not be determined
D31 Hospital
D32 Halfway house
D33 Nursing home, retirement home, or home for the aged
D34 County home or poor farm
D35 Orphanage
D36 Jail or detention center
<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>D37</td>
<td>Federal penitentiary, State prison, or prison farm</td>
</tr>
<tr>
<td>D40</td>
<td>Educational or religious institution, major category</td>
</tr>
<tr>
<td>D41</td>
<td>Sorority or fraternity</td>
</tr>
<tr>
<td>D42</td>
<td>Convent or monastery</td>
</tr>
<tr>
<td>D43</td>
<td>Educational institution, including academy, school, college, and university</td>
</tr>
<tr>
<td>D44</td>
<td>Religious institution, including church, synagogue, seminary, temple, and mosque</td>
</tr>
<tr>
<td>D50</td>
<td>Transportation terminal; major category used alone when the minor category could not be determined</td>
</tr>
<tr>
<td>D51</td>
<td>Airport or airfield</td>
</tr>
<tr>
<td>D52</td>
<td>Train station</td>
</tr>
<tr>
<td>D53</td>
<td>Bus terminal</td>
</tr>
<tr>
<td>D54</td>
<td>Marine terminal</td>
</tr>
<tr>
<td>D55</td>
<td>Seaplane anchorage</td>
</tr>
<tr>
<td>D56</td>
<td>Subway or metro station</td>
</tr>
<tr>
<td>D57</td>
<td>Parking Area Boundary</td>
</tr>
<tr>
<td>D58</td>
<td>TANA/GDT airport property boundary</td>
</tr>
<tr>
<td>D59</td>
<td>TANA/GDT airport runway</td>
</tr>
<tr>
<td>D60</td>
<td>Employment center; major category used alone when the minor category could not be determined</td>
</tr>
<tr>
<td>D61</td>
<td>Shopping center or major retail center</td>
</tr>
<tr>
<td>D62</td>
<td>Industrial building or industrial park</td>
</tr>
<tr>
<td>D63</td>
<td>Office building or office park</td>
</tr>
<tr>
<td>D64</td>
<td>Amusement center</td>
</tr>
<tr>
<td>D65</td>
<td>Government center</td>
</tr>
<tr>
<td>D66</td>
<td>Other employment center</td>
</tr>
<tr>
<td>D67</td>
<td>Stadium</td>
</tr>
<tr>
<td>D70</td>
<td>Tower; major category used alone when minor category could not be determined</td>
</tr>
<tr>
<td>D71</td>
<td>Lookout tower</td>
</tr>
<tr>
<td>D80</td>
<td>Open space; major category used alone when the minor category could not be determined</td>
</tr>
<tr>
<td>D81</td>
<td>Golf course</td>
</tr>
<tr>
<td>D82</td>
<td>Cemetery</td>
</tr>
<tr>
<td>D83</td>
<td>National Park Service land</td>
</tr>
<tr>
<td>D84</td>
<td>National forest or other Federal land</td>
</tr>
<tr>
<td>D85</td>
<td>State or local park or forest</td>
</tr>
<tr>
<td>D89</td>
<td>Local Park or Recreation Area</td>
</tr>
</tbody>
</table>
D90  Special purpose landmark; major category used alone when the
       minor category could not be determined
D91  Post office box ZIP Code(R)
D92  Urbanizacion, an identifiable community development in Puerto
       Rico
D93  Fire Department
D94  Police Station
D95  Library
D96  City/Town Hall
E00  Physical feature, tangible but not transportation or hydrographic;
       major and minor categories unknown
E10  Fence line locating a visible and permanent fence between
       separately identified property
E20  Topographic feature; major category used when the minor category
       could not be determined
E21  Ridge line, the line of highest elevation of a linear mountain
E22  Mountain peak, the point of highest elevation of a mountain
E23  Island, identified by name
E24  Levee, an embankment, as of earth or concrete, used to prevent a
       river or other body of water from overflowing
F00  Nonvisible boundary, major and minor categories unknown
F10  Nonvisible jurisdictional boundary of a legal or administrative entity,
       major category
F11  Offset boundary of a legal or administrative entity
F12  Corridor boundary of a legal or administrative entity
F13  Nonvisible superseded 2000 governmental unit boundary
F14  Superseded 1990 legal boundary
F15  Superseded 1990 legal boundary, corrected through post census
       process
F16  Superseded legal boundary, current at the time of the 1997
       Economic Census
F17  Nonvisible State Legislative District boundary
F18  Nonvisible Congressional District boundary
F19  Nonvisible corrected 2000 governmental unit boundary
F20  Nonvisible feature for data base topology; major category used
       when the minor category could not be determined
F21  Automated feature extension to lengthen existing physical feature
F22  Irregular feature extension, determined manually, to lengthen
       existing physical feature
F23  Closure extension to complete data base topological closure
       between extremely close features
F24  Nonvisible separation line used with offset and corridor boundaries
<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F25</td>
<td>Nonvisible centerline of area enclosed by corridor boundary</td>
</tr>
<tr>
<td>F30</td>
<td>Point-to-point line, follows a line of sight and should not cross any visible feature</td>
</tr>
<tr>
<td>F40</td>
<td>Property line, nonvisible boundary of either public or private lands, e.g., a park boundary</td>
</tr>
<tr>
<td>F50</td>
<td>ZIP Code(R) tabulation boundary, used in delineating ZIP Code(R) Tabulation Areas</td>
</tr>
<tr>
<td>F60</td>
<td>Map edge, now removed, used during data base creation</td>
</tr>
<tr>
<td>F70</td>
<td>Statistical boundary; major category used when the minor category could not be determined</td>
</tr>
<tr>
<td>F71</td>
<td>1980 statistical boundary</td>
</tr>
<tr>
<td>F72</td>
<td>1990 statistical boundary, used to hold collection and tabulation census block boundaries not represented by existing physical features</td>
</tr>
<tr>
<td>F73</td>
<td>1990 statistical boundary and extent of land use, it is not classifiable as a physical feature</td>
</tr>
<tr>
<td>F74</td>
<td>1990 statistical boundary, used to hold a tabulation census block boundary not represented by an existing physical feature</td>
</tr>
<tr>
<td>F75</td>
<td>1990 statistical boundary and extent of land use, it is not classifiable as a physical feature</td>
</tr>
<tr>
<td>F76</td>
<td>1990 statistical boundary, used to hold a tabulation census block boundary not represented by an existing physical feature</td>
</tr>
<tr>
<td>F80</td>
<td>Nonvisible other tabulation boundary, major category used when the minor category could not be determined</td>
</tr>
<tr>
<td>F81</td>
<td>School district tabulation boundary</td>
</tr>
<tr>
<td>F82</td>
<td>Special census tabulation boundary</td>
</tr>
<tr>
<td>F83</td>
<td>Census 2000 collection block boundary; used to hold Census collection block boundaries not represented by existing physical features</td>
</tr>
<tr>
<td>F84</td>
<td>Census 2000 statistical area boundary; used to hold Census statistical area boundaries not represented by existing physical features</td>
</tr>
<tr>
<td>F85</td>
<td>Census 2000 tabulation block boundary; used to hold Census tabulation block boundaries not represented by existing physical features</td>
</tr>
<tr>
<td>F86</td>
<td>Internal U.S. Census Bureau use</td>
</tr>
<tr>
<td>F87</td>
<td>Oregon urban growth area boundary</td>
</tr>
<tr>
<td>F88</td>
<td>Current statistical area boundary</td>
</tr>
<tr>
<td>H00</td>
<td>Water feature, classification unknown or not elsewhere classified</td>
</tr>
<tr>
<td>H01</td>
<td>Shoreline of perennial water feature</td>
</tr>
<tr>
<td>H02</td>
<td>Shoreline of intermittent water feature</td>
</tr>
<tr>
<td>H10</td>
<td>Stream boundary, major category</td>
</tr>
<tr>
<td>H11</td>
<td>Perennial stream or river</td>
</tr>
<tr>
<td>H12</td>
<td>Intermittent stream, river, or wash</td>
</tr>
<tr>
<td>H13</td>
<td>Braided stream or river</td>
</tr>
<tr>
<td>H20</td>
<td>Canal, ditch, or aqueduct, major category</td>
</tr>
<tr>
<td>H21</td>
<td>Perennial canal, ditch, or aqueduct</td>
</tr>
<tr>
<td>H22</td>
<td>Intermittent canal, ditch, or aqueduct</td>
</tr>
</tbody>
</table>
H30  Lake or pond; major category used when the minor category could not be determined
H31  Perennial lake or pond
H32  Intermittent lake or pond
H40  Reservoir; major category used when the minor category could not be determined
H41  Perennial reservoir
H42  Intermittent reservoir
H50  Bay, estuary, gulf, sound, sea, or ocean; major category used when the minor category could not be determined
H51  Bay, estuary, gulf, or sound
H53  Sea or ocean
H59  TANA/GDT Shoreline
H60  Gravel pit or quarry filled with water
H70  Nonvisible water area definition boundary; used to separate named water areas; major category
H71  USGS closure line, used as maritime shoreline
H72  Census water centerline, computed to use as median positional boundary
H73  Census water boundary, international in waterways or at 12-mile limit, used as area measurement line
H74  Census water boundary, separates inland from coastal or Great Lakes, used as area measurement line
H75  Census water boundary, separates coastal from territorial sea at 3-mile limit, used as area measurement line
H80  Special water feature, major category used when the minor category could not be determined
H81  Glacier
X00  Feature not yet classified

See Also

Geocoding Data Sources
Import a Drawing from the Geocoding Database
Street Address Geocoding
Import Drawing - Geodatabase

Drawings may be imported or linked from ESRI geodatabase files, either SDE geodatabases or Personal geodatabases. Geodatabase files save object metric information within database tables using binary data types.

Personal geodatabases normally use .mdb files. Use Database Console to connect (via the Data Source dialog) to the .mdb file hosting the Personal Geodatabase and then Manifold's spatial DBMS capability will automatically identify the contents of that geodatabase. We can then import or link drawings as desired. See the spatial DBMS topic.

SDE geodatabases can use several different DBMS packages, but are usually found hosted in SQL Server or Oracle. When Database Console connects to such a data source it will recognized SDE geodatabases when it sees them. We can then import or link drawings as desired.

Important Note when Using 64-bit Manifold Editions

Due to a lack of required Microsoft facilities in 64-bit Windows systems, Manifold in 64-bit mode cannot export, import, export or link to DB, HTML, MDB, XLS or WKx format files. This includes no access to the MDB parts of Manifold MFD and MapInfo TAB imports, or an MDB used to store an ESRI geodatabase. The workaround for importing or exporting such files is to launch Manifold in 32-bit mode by using the Manifold System (32-bit) shortcut, perform the export from or import into a .map project file and then re-launch Manifold in 64-bit mode using the Manifold System (64-bit) shortcut. Linking is more complex: the data must be in some format usable in 64-bit mode within 64-bit Windows systems, such as a SQL Server database. Alternatively, the data can be kept within a Manifold .map project file and linked using the Manifold ODBC driver.

See Also

Geometry in Tables
Linked Drawings
Queries and Geoms
Import Drawing - GML

GML is an abbreviation for "Geography Markup Language." A spectacularly inefficient format, GML is based on XML and is being touted as a proposed standard in some circles. Unfortunately, GML is something of a non-standard since every implementation of GML to date for saving GIS data has been incompatible with other implementations. Besides the intrinsic incompatibility designed into GML, the main problem with GML is the extreme inefficiency of the format: GML will frequently require files that are over 100 megabytes in size to store GIS data that other formats can save in only five megabytes.

GML is currently used by the UK's Ordnance Survey (OS) for publishing MasterMap data products, the only concrete form of GML that is currently used to publish a significant amount of data. The Manifold GML importer is therefore an importer for Ordnance Survey GML format, specifically, for OS MasterMap data products. manifold.net expects to work closely with the OS to assure increasing compatibility with OS usage of GML.

GML files may have a .gml extension or, as is the case with some Ordnance Survey (OS) data files with no extension. GML files can occur in both independent and topological polygon forms. Manifold will automatically import either type and will apply any updates specified. Manifold's GML importer supports object versions and will import any metadata as a Manifold comments component.

One interesting aspect of GML format is that a .gml files can contain a complete version of a drawing or it can contain updates to a drawing that represent changes to be made. In fact, a .gml file can contain both an original version as well as subsequent updates.

Importing a GML File

1. Choose File - Import - Drawing
2. In the Import dialog choose GML files in the Files of type box and browse to the folder containing the GML file to be imported. Double click on the file.
3. The Import GML Data Set dialog will pop open to allow specification of any update files to be processed as part of the import. If importing UK Ordnance Survey MasterMap files, make sure the MasterMap data box is checked. It is checked by default since MasterMap is the only known significant collection of files in GML format.
4. If one or more update files are to be processed, click on the New button in the dialog's toolbar and in the resulting Append dialog double click on the desired update file. Using the New button, add as many update files are desired. The update files will be processed in order from top to bottom, so arrange the list in the order desired using the Move buttons in the dialog's toolbar. Press OK.

Example

We will import one of the MasterMap samples provided by the OS on their website at www.ordnancesurvey.co.uk. The example is a map of Birmingham that is provided as an "initial data" file and as a "change-only" update file. The initial data file is called 101-SP0482-2c1.gz (5.6 MB in size) and the update file is called 102-SP0482-2c2.gz (475 KB in size). Given the similarity of names it makes sense for us to avoid confusing the two by keeping the files in different folders, perhaps called "Birmingham initial" and "Birmingham Update."

<table>
<thead>
<tr>
<th>Name</th>
<th>Size (MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>101-SP0482-2c1.gz</td>
<td>5.649</td>
</tr>
<tr>
<td>102-SP0482-2c2.gz</td>
<td>0.475</td>
</tr>
</tbody>
</table>

The .gz extension indicates the files have been zipped with GnuZIP, which is easily unzipped using WinZip. We routinely use WinZip to unzip .gz files, so our Windows explorer shows the WinZip icon next to the .gz file as seen above.
When WinZip unzips the files it asks us what extension they should have since the OS apparently does not use extensions for MasterMap files.

We used the extension .gml to indicate that it is a GML file so if we encounter these files in the future we won't wonder what they are.

Upon decompression the initial data file, now named 101-SP0482-2c1.gml, has expanded to 105 MB (!), while the update file, named 102-SP0482-2c2.gml, has expanded to 5.9 MB. Note that it would be more efficient to send changes by repeating the entire file in a more efficient GIS format (which would be about the size of the 5.6 MB zipped original file) than it is to use an unzipped GML changes file.

We repeat the procedure above to unzip the update file, which will now be called 102-SP0482-2c2.gml.
To import the .gml file into Manifold we launch Manifold System and then choose File - Import - Drawing using GML Files in the Files of type box. We browse over to our initial data .gml file and choose it.

Manifold opens an Import GML Data Set dialog to allow us to specify any update files desired. We click on the New button to add a new update.

This opens up an Append dialog that allows us to find and add another .gml file as an update file. We browse over to the folder holding our update file and double click on it.
The Import GML Data Set dialog is now loaded with the update file we desire. If we click on the update file to highlight it in the Updates pane the other toolbar buttons will be enabled, so we could delete this update or move it up and down in order relative to other update files we might add to the pane. Press OK and the file will import.

GML format is so inefficient that it can take a very long time to import a GML file, perhaps 20 minutes on an average computer to import a 100 MB GML file. Have patience. Eventually, the result is a drawing and some other components in the project pane.

If we open the drawing and apply some thematic formatting to area background color (using the theme field) we can see the excellent detail in the OS MasterMap product. Effective use of MasterMap requires some familiarity with the Ordnance Survey’s data organization plan. See the OS’s web site for documentation.

Import GML Data Set Dialog Controls
New - Add a new update file.

Move to Top - Move the highlighted update file to the top of the update file stack. The top file will be the first from which updates are applied.

Move Up - Move the highlighted update file up one position in the update file stack.

Move Down - Move the highlighted update file down one position in the update file stack.

Move to Bottom - Move the highlighted update file to the bottom of the update file stack. The bottom file will be the last from which updates are applied.

Delete - Delete highlighted update file.

MasterMap data
Checked by default. When checked, parses the GML stream for projection information used in UK Ordnance Survey Master Map files.

Versions
GML entities include version information that allows distinguishing between multiple versions of the same object (that is, between objects with same unique ID assigned by the data provider). When Manifold encounters an object that has already been imported, perhaps as the result of processing an update, it compares the versions of the existing object and the new object, determines which object is more recent and uses the more recent object.

Notes
The capability to receive updates was presumably motivated by the immense file sizes required by GML. With updates, the original large file would not have to be downloaded or copied when changes need to be made: a smaller GML file containing changes could be provided so that the original file could be used in conjunction with the smaller changes file. However, GML is so inefficient that just about any other GIS format can contain a refreshed version of the entire data set and still be smaller than only the changes expressed in GML. Manifold does not at present support use of incremental change files, but this capability is expected to be added in future releases of Manifold.

One way to get around the large file sizes imposed by GML is to do what the Ordnance Survey, does, which is to provide files already zipped in .gz format. The OS suggests that applications might read directly from the zipped file format, but that is not always a good way of dealing with the inefficiencies of GML.

Because GML files take a very long time to import, if Manifold imported directly from a .gz file in the case of a damaged file users could spend a long time on import finding out that the file was damaged in download or copying. If the .gz file is unzipped first (a rapid process) any errors in the file will be discovered right away before the long process of GML import begins.

Note that the 105 MB Birmingham sample file in GML format takes tens of minutes to import into Manifold. However, it can be saved as a .map project file requiring only 5.8 MB of space in a few tens of seconds and thereafter loaded even faster. Compared to efficient GIS formats, GML format is not only 20 times larger, it is also about 60 times slower. For this reason, we recommend that Manifold users import GML files and then save them in more efficient formats, either as Manifold projects or, for large data, within spatial DBMS storage.

For all the limitations of GML there is much UK data currently available in OS GML form and it is likely that this format will continue in use by the Ordnance Survey. To support UK users of Manifold, manifold.net expects to work closely with the OS to assure increasing compatibility with OS usage of GML.

UK users of Ordnance Survey data in GML format are encouraged to send suggestions to manifold.net for improvements and conveniences they would like to see within enhanced support for GML import by Manifold. See the Contacting manifold.net topic for information on contributing suggestions.
Import Drawing - KML, KMZ

KML is an XML-based format originally used for "annotations" in Google Earth displays. KMZ is exactly the same format compressed using "zip" compression with a three-letter file name extension of .kmz. KML can be used for points, lines or areas with up to two text attributes, a Name field and a Description field, per object. Manifold includes the ability to import a drawing from a Google Earth KML or KMZ file. If a KML file contains more than one folder, each folder will import as a separate drawing.

KML and KMZ provide data in Latitude / Longitude projection using the WGS 84 Auto datum only.

Notes

The initial KML importer in 7x is a simple importer that grabs vector data for points, lines and areas plus the Name and Description values, if present. It does not attempt to translate Google styles or colors into Manifold styles or colors.

Manifold can also export drawings and images to KML and KMZ. See the Export Drawing - KML, KMZ and Export Image - KML, KMZ topics for additional import information.

See Also

A Flashy Demo - Web Queries and KML
Exporting KML to Google Earth
Fun with Google Earth
Linked Images from Google Servers
Export Drawing - KML, KMZ
Export Image - KML, KMZ
Import Drawing - LULC

LULC (Land Use and Land Cover) maps are created by USGS for use with their GIRAS information system. The maps are matched to 1:250,000-scale base maps and (in some areas of the US) a limited number of 1:100,000-scale base maps. Land use and land cover data provides information on urban or built up land, agricultural land, rangeland, forest land, water, wetlands, barren land, tundra, and perennial snow or ice. Associated maps display information in five data categories: (1) political units, (2) hydrologic units, (3) census county subdivisions, (4) Federal land ownership, and (5) State land ownership.

The screen shot above shows the San Francisco Bay region. Orange-yellow indicates inland waters (bays, etc) while bright blue indicates ocean. Other color values were chosen using the Temperature preset using Unique Values in the thematic formatting dialog.

The main source for LULC maps are free downloads via Internet from the USGS geodata download page at http://edc.usgs.gov/doc/edchome/ndcdb/ndcdb.html (if this URL changes, start with www.usgs.gov and drill down from their to find the LULC download area).

For each 1:250K or 1:100K region served by USGS, several files showing land use, hydrological regions, political boundaries, census county subdivisions and so forth may be available.

Downloaded files will appear with a .gz extension, since they are compressed with gnu-zip UNIX style compression. Use WinZip or any other .gz-capable decompression utility to unzip them. LULC files do not have a three letter extension by default. We recommend using .luc or similar extension as a mnemonic. Import them using File - Import - Drawing using LULC GIRAS Files (*.*) as the Files of Type choice in the Import dialog.

See the USGS pages for detailed documentation describing all the different data codes used to classify different land use regions in LULC.

Classification Codes

Land use / Land cover classification Codes-first and second level categories:

1 Urban or Built-Up Land
   11 Residential
   12 Commercial Services
   13 Industrial
   14 Transportation, Communications
   15 Industrial and Commercial
16 Mixed Urban or Built-Up Land
17 Other Urban or Built-Up Land

2 Agricultural Land
21 Cropland and Pasture
22 Orchards, Groves, Vineyards, Nurseries
23 Confined Feeding Operations
24 Other Agricultural Land

3 Rangeland
31 Herbaceous Rangeland
32 Shrub and Brush Rangeland
33 Mixed Rangeland

4 Forest Land
41 Deciduous Forest Land
42 Evergreen Forest Land
43 Mixed Forest Land

5 Water
51 Streams and Canals
52 Lakes
53 Reservoirs
54 Bays and Estuaries

6 Wetland
61 Forested Wetlands
62 Non-forested Wetlands

7 Barren Land
71 Dry Salt Flats
72 Beaches
73 Sandy Areas Other than Beaches
74 Bare Exposed Rock
75 Strip Mines, Quarries, and Gravel Pits
76 Transitional Areas
77 Mixed Barren Land

8 Tundra
81 Shrub and Brush Tundra
82 Herbaceous Tundra
83 Bare Ground
84 Wet Tundra
85 Mixed Tundra

9 Perennial Snow and Ice

91 Perennial Snowfields

92 Glaciers

**Notes**

GIRAS stands for Geographic Information Retrieval Analysis System.

Oceans may be coded with a value of 2000000102.
Import Drawing - ETAK MapBase
A proprietary format used to import ETAK GIS data sets. ETAK has been acquired by TeleAtlas and in the future may be known as TeleAtlas only.

Select All - Check all drawings. All drawings will be imported with this drawing.

Select None - Uncheck all drawings. No data drawings will be imported. Tech tip: Use this to uncheck all drawings before checking those desired.

Select Inverse - Uncheck all previously checked drawings and check all previously unchecked drawings. This is a fast way to check only a few drawings out of many: uncheck those desired and then push Select Inverse.

Drawings pane
Check the drawings to be imported. Drawings typically group areas, lines and points as well as other features such as bridges or tunnels, turn restrictions and unmapped address ranges.

See Also
Import and Export

**Import Drawing - HTML**

Manifold can create a drawing directly from data stored in a geocoded table. This capability is directly analogous to the creation of a linked drawing from a geocoded table except that instead of the drawing being linked to the originating table it is created as an independent drawing within the Manifold project.

Such imports are accomplished by choosing **File - Import - Drawing** and then choosing **Data Sources** () in the **Files of type** box to launch the Data Source dialog. The desired data source can then be configured and used in the Data Source dialog.

Drawings can be imported from geocoded tables using any of the data access methods, such as from an HTML file, provided by Manifold. Dialogs used are the same as for creating a linked drawing from a geocoded table, except that linked drawings always invoke the Data Source dialog and the result of an import is an imported drawing rather than a linked drawing.

Note that what can visually appear to be a "table" in an HTML file to the eye might not really be organized as a table from the perspective of Microsoft's table-reading, data access software. If Access cannot find a table to read in the HTML file, Manifold will not be able to either.

**Important Note when Using 64-bit Manifold Editions**

Due to a lack of required Microsoft facilities in 64-bit Windows systems, Manifold in 64-bit mode cannot export, import, export or link to DB, HTML, MDB, XLS or WKx format files. This includes no access to the **MDB** parts of Manifold MFD and MapInfo TAB imports. The workaround for importing or exporting such files is to launch Manifold in 32-bit mode by using the **Manifold System (32-bit)** shortcut, perform the export from or import into a .map project file and then re-launch Manifold in 64-bit mode using the **Manifold System (64-bit)** shortcut. Linking is more complex: the data must be in some format usable in 64-bit mode within 64-bit Windows systems, such as a SQL Server database. Alternatively, the data can be kept within a Manifold .map project file and linked using the Manifold ODBC driver.

Although Microsoft is slowly moving forward towards 64-bit operation of Office and the JET components used by Access and Excel, at the present writing (2010) this is such a total kludge that while it is possible to get 64-bit access to Access or Excel it is just easier for most to continue using the 32-bit versions. See comments in the 32-bit and 64-bit Manifold Editions topic.

**See Also**

The Data Source Dialog
Importing and Linking Tables
Linked Drawings
Create a Linked Drawing from a Geocoded Table
Import Drawing - MDB

Manifold can create a drawing directly from data stored in a geocoded table. This capability is directly analogous to the creation of a linked drawing from a geocoded table except that instead of the drawing being linked to the originating table it is created as an independent drawing within the Manifold project.

Such imports are accomplished by choosing File - Import - Drawing and then choosing Data Sources () in the Files of type box to launch the Data Source dialog. The desired data source can then be configured and used in the Data Source dialog.

Drawings can be imported from geocoded tables using any of the data access methods, such as from a Microsoft Access MDB file, provided by Manifold. Dialogs used are the same as for creating a linked drawing from a geocoded table, except that linked drawings always invoke the Data Source dialog and the result of an import is an imported drawing rather than a linked drawing.

Important Note when Using 64-bit Manifold Editions

Due to a lack of required Microsoft facilities in 64-bit Windows systems, Manifold in 64-bit mode cannot export, import, export or link to DB, HTML, MDB, XLS or WKx format files. This includes no access to the MDB parts of Manifold MFD and MapInfo TAB imports. The workaround for importing or exporting such files is to launch Manifold in 32-bit mode by using the Manifold System (32-bit) shortcut, perform the export from or import into a .map project file and then re-launch Manifold in 64-bit mode using the Manifold System (64-bit) shortcut.

Linking is more complex: the data must be in some format usable in 64-bit mode within 64-bit Windows systems, such as a SQL Server database. Alternatively, the data can be kept within a Manifold .map project file and linked using the Manifold ODBC driver.

Although Microsoft is slowly moving forward towards 64-bit operation of Office and the JET components used by Access and Excel, at the present writing (2010) this is such a total kludge that while it is possible to get 64-bit access to Access or Excel it is just easier for most to continue using the 32-bit versions. See comments in the 32-bit and 64-bit Manifold Editions topic.

See Also

The Data Source Dialog
Importing and Linking Tables
Linked Drawings
Create a Linked Drawing from a Geocoded Table
Import Drawing - MFD
Import drawings from Manifold System Release 4.50 .mfd/.mdb format. The importer will work even if the .mfd is not accompanied by any .mdb file, in which case it will import drawing objects only without any data attribute fields in the associated table.

Select All - Check all fields. All data fields will be imported with this drawing.

Select None - Uncheck all fields. No data fields will be imported. Tech tip: Use this to uncheck all fields before checking those desired.

Select Inverse - Uncheck all previously checked fields and check all previously unchecked fields. This is a fast way to check only a few columns out of many: uncheck those desired and then push Select Inverse.

Fields pane Check the data attribute fields to be imported.

Import by layers Enabled if the .mfd has embedded layers. Check to import each layer as a separate drawing.

Layers in Release 4.50 workspaces are normally saved in the .mws workspace file, which references geometry stored in .mfd files and tables stored in .mdb files. .mfd files can also contain a layering structure. This is not normally used except to prepare .mfd/.mdb files for publication with a pre-built layer structure. If the .mfd file does have embedded layers the Import by layers checkbox will be enabled.

Comments
Technically, there was not an absolute requirement in Release 4.50 to name the .mfd and .mdb with similar names, that is, the same filename before the three letter .mfd or .mdb extension. The MFD importers in 5.00 and later versions of Manifold, however, require that the .mdb file is named the same as the .mfd file.

Important Note when Using 64-bit Manifold Editions
Due to a lack of required Microsoft facilities in 64-bit Windows systems, Manifold in 64-bit mode cannot export, import, export or link to DB, HTML, MDB, XLS or WKx format files. This includes no access to the MDB parts of Manifold MFD and MapInfo TAB imports. The workaround for importing or exporting such files is to launch Manifold in 32-bit mode by using the Manifold System (32-bit) shortcut, perform the export from or import into a .map project file and then re-launch Manifold in 64-bit mode using the Manifold System (64-bit) shortcut. Linking is more complex: the data must be in some format usable in 64-bit mode within 64-bit Windows systems, such as a SQL Server database. Alternatively, the data can be kept within a Manifold .map project file and linked using the Manifold ODBC driver.

Although Microsoft is slowly moving forward towards 64-bit operation of Office and the JET components used by Access and Excel and the MDB part of Manifold 4.50 MFD/MDB format, at the present writing (2010) this is such a total kludge that while it is possible to get 64-bit access to Access or Excel it is just easier for most to continue using the 32-bit versions. See comments in the 32-bit and 64-bit Manifold Editions topic.

See Also
Import Drawing - MWS for information on importing Manifold System 4.50 .mws workstation files.
Import Drawing - MIF

MapInfo .mid/.mif format is widely used for interchange with MapInfo and other GIS applications. The format normally uses two files that have the same base name but with a .mif or .mid extension. The .mif file stores geometry information and the .mid file stores data attributes (fields). If a .mid file is present Manifold will import the table of data attributes it contains.

Select All - Check all fields. All data fields will be imported with this drawing.

Select None - Uncheck all fields. No data fields will be imported. Tech tip: Use this to uncheck all fields before checking those desired.

Select Inverse - Uncheck all previously checked fields and check all previously unchecked fields. This is a fast way to check only a few columns out of many: uncheck those desired and then push Select Inverse.

Fields pane Check the data attribute fields to be imported.

Import formatting When checked (default) imports colors and styles used for areas, area borders, lines and points. Imported formatting will be implemented using thematic formatting if the drawing includes two or more objects with different formatting values.

Formatting

Mif files can contain color information as well as style parameters for objects. Manifold will read foreground color, background color and style information for objects on import, including formatting for area borders. When exporting to mif format, Manifold will save foreground color, background color and size as well. If no color or size information is present in the file, Manifold will use default format settings.

Because sizes are not the same in MapInfo and Manifold, the .mif importer will attempt match sizes as best as possible.

Data Types

On import of a MapInfo .mif or .tab file, Manifold will translate MapInfo logical data type columns into Boolean data type. Exporting a Manifold drawing to .mif will export Boolean columns into MapInfo logical data type columns.

Datum Naming

Formats such as MapInfo .mif and .tab do not actually store the names of datums within the projection information stored in the file; instead, the numeric parameters for the datum are saved. When Manifold scans such a file the system will examine these numeric parameters to find a named datum that is a match and will use that datum's name to identify the datum in use.

In some cases, more than one datum may use the same numeric parameters. For example, there are many national datums that are mathematically identical to some generically named datums. When several datums match the given numeric parameters, Manifold will choose a neutrally named datum in preference to a name that uses a specific country name.

See Also

Import Drawing - TAB for information on importing MapInfo TAB format files.
Import Drawing - MWS

Manifold System Release 4.50 saves drawings using two files. An .mfd file saves geometry and a similarly-named .mfd file saves data attributes (fields). A third file, an .mws workstation file, saves the 4.50 workspace. The .mws file contains pointers to the various .mfd/.mdb files that make up a given workspace and contains information on how objects should be formatted, labels, layer structure and other information.

When importing .mws files from Release 4.50, one must therefore have not only the .mws file but also all of the .mfd/.mdb files it references. The .mws importer will import each .mfd/.mdb referenced as a separate drawing and will attempt to preserve imports.

Formatting

Because of differences between formats in Release 4.50 and subsequent Manifold editions, not all formatting information stored in an MWS file can be imported automatically into current Manifold editions. A format applied throughout an entire .mfd file will in most cases be imported, but "per object" formats will not. In addition, 4.50 styles for areas, lines and points are different from styles in subsequent Manifold editions. Labels are also different.

The importer will attempt to provide reasonable approximations; however, manual adjustment of formats will usually be required.

Manual Retrieval of Layers

Although the .mws importer will usually be able to get layers out of the .mws, we may be required in rare cases to retrieve layer structures manually. Suppose we wish to transition a counties.mfd file that was shown in four layers in a 4.50 workspace. We import the .mfd to create a countiesDrawing and a countiesTable in the project. We open the countiesDrawing and then while it is open we use File - Create - Drawing to create four new (empty) drawings in the project. Let's call these DrawingA, DrawingB, DrawingC and DrawingD.

We can now select the various items from countiesDrawing that we would like in different layers. We select each layer's worth of objects and use CTRL-X or Edit - Cut to cut them out of countiesDrawing, open the target drawing (A, B or whatever) and Paste the objects into the target drawing. To move objects between drawings we use Cut and Paste.

Once all of the desired objects have been cut out of countiesDrawing and pasted into the target drawings we can assemble all of the drawings into a map. This is easy: use File - Create - Map to create a map and check the boxes for the drawings that are to appear as layers in the map.

See Also

Import Drawing - MFD for information on importing Manifold System Release 4.50 .mfd/.mdb files.
Import Drawing - NTAD

NTAD format is used to import drawings from the U.S. Bureau of Transportation Statistics (BTS) National Transportation Atlas Databases (NTAD) and the BTS North American Transportation Atlas Data (NORTAD) CDs. BTS (www.bts.gov) provides fabulous, first rate data on various transportation facilities, road and rail networks and other useful data layers such as counties, states, national park boundaries, etc. At the present writing, BTS will send CDs at no charge when ordered from their web site.

NTAD files are usually organized into files that contain areas (.are), lines (.lnk) or points (.nod or .pnt). Import the desired files by browsing to the desired files.

Select All - Check all fields. All data fields will be imported with this drawing.

Select None - Uncheck all fields. No data fields will be imported. Tech tip: Use this to uncheck all fields before checking those desired.

Select Inverse - Uncheck all previously checked fields and check all previously unchecked fields. This is a fast way to check only a few columns out of many: uncheck those desired and then push Select Inverse.

Fields pane
Check the data attribute fields to be imported.

Map fields between tables
Create relations within drawing tables to link to common tables.

A Note on .lnk Files

Because Windows uses files with extension .lnk to indicate a link, files using this extension will not be shown in the Import dialog with the extension.

Instead, they will be seen with a small "link" sub-icon in the corner of the file’s icon. The illustration shows a .lnk and a .nod file that make up the National Highway Planning Network (NHPN), a road network of essentially all roads used for intercity transportation in the United States.

Historical Note

Manifold’s first use as a GIS in version 2.00 was to display the NHPN. Developed at Oak Ridge National Laboratory for the U.S. Army, the NHPN was generously provided to manifold.net on a SUN tape cartridge.
Import Drawing - NTF

The Ordnance Survey (OS) is the state cartographic monopoly in the United Kingdom. NTF format is used by the OS to publish both drawings in the Land-Line series as well as surfaces in their DTM grid series. Manifold’s NTF importer will automatically create drawings or surfaces depending on the contents of the NTF.

- **Select All** - Check all fields. All data fields will be imported with this drawing.

- **Select None** - Uncheck all fields. No data fields will be imported. Tech tip: Use this to uncheck all fields before checking those desired.

- **Select Inverse** - Uncheck all previously checked fields and check all previously unchecked fields. This is a fast way to check only a few columns out of many: uncheck those desired and then push **Select Inverse**.

**Fields pane** Check the data attribute fields to be imported. (Not applicable when importing a surface).
Import Drawing - SDTS

Spatial Data Transfer Standard (SDTS) format is emerging as the primary GIS data interchange standard for the U.S. federal government. Most USGS drawings (DLGs) and surfaces (DEMs) are now provided using SDTS format. SDTS format is used for both drawings and surfaces. Manifold will automatically create the correct component depending on the contents of the SDTS files.

SDTS format normally includes a large number of files organized within a directory. One of the files is a catalog file that ends in `*.CATD.DDF`. To import an SDTS data set, browse to the directory that contains the files and import the `CATD.DDF` file. Manifold will read the catalog and automatically organize the import of all files involved.

Select All - Check all modules. All data modules will be imported with this drawing.

Select None - Uncheck all modules. No data modules will be imported. Tech tip: Use this to uncheck all modules before checking those desired.

Select Inverse - Uncheck all previously checked modules and check all previously unchecked modules. This is a fast way to check only a few columns out of many; uncheck those desired and then push Select Inverse.

Modules pane
- Check the modules to be imported. Modules will often be organized as containing points, lines or areas.

Map fields between tables
- Create relations within drawing tables to link to common tables. (Not applicable when importing a surface).

Import data definition scheme
- Create an additional table that provides the data definition scheme used for modules in other tables. (Not applicable when importing a surface).

Import Z
- If Z data is available, create an additional drawing with points that provide a Z coordinate.

Use long module names
- Checked by default. Create names for imported component using the TITLE field stored in the identity module. If not checked, creates a shorter name based on the name of the .ddf module.

Use master data dictionary
- Read the specified SDTS master data dictionary. Use the … browse button to navigate to the master data dictionary desired.

SDTS format allows objects such as lines to be defined not just by X,Y coordinates but also by Z coordinates at each coordinate location that defines the line. If Z data exists in the data set, checking the Import Z box causes an additional drawing to be created filled with points, each of which has a Z coordinate taken from the accessory Z values in the coordinate sequences that define lines or other objects. This drawing can then be used to create a surface.

Long names will be prefixed with the module identifier. This allows distinct names like "Hy01 Drawing" (hydrography or hypsography) and "Bd01 Drawing" (boundaries) even when the SDTS identification module contains no titles.

Importing an SDTS file can create many files: drawings, comments and possibly several tables can be created. To keep the project organized, create a folder into which each SDTS data set can be imported.

The Map fields between tables option if checked will automatically create relations between the drawing's table and other tables. Since potentially very many tables can be imported, the relations will not be used within the drawing's table until we choose the columns we want to be used.

To use a relation built during SDTS file import:

1. Open the drawing's table.
2. Launch Table - Relations.
3. Click the desired relation to select it.
4. Check the columns to be included in the drawing's table from that relation. Click OK.

**Tech Tip**

Some SDTS files published by the US government for regions in Puerto Rico contain an error in that the data they contain is projected using the Puerto Rico datum but the SDTS file information says the contents use the NAD27 datum. When such drawings are imported and overlaid on TIGER/Line 2000 drawings they will not align with the TIGER/Line data. This problem has been observed in 1:24K-scale SDTS DLG files as well as 1:24K-scale SDTS DEM files. It may occur in other files for Puerto Rico that are in SDTS format as well.

To fix this problem, open the drawing and use the Edit - Assign Projection dialog to change the datum to the Puerto Rico datum.
Import Drawing - SHP, Shapefiles

ESRI's .shp format, also known as "shape format" or "shapefiles," is used with ArcView, a popular GIS package of the early 1990's. Shape format has been openly published by ESRI and is widely used for data interchange in GIS. Shapefiles usually consist of three similarly named files with differing extensions: a .shp, .shx and a .dbf file. The .dbf file is a dBase database system format file that is used to store data attributes for the drawing. Manifold's importer will import .dbf attributes if present and will import .shp geometry information without data attributes if no .dbf is present.

Shapefiles may also include a .prj file that purports to give coordinate system (projection) information. Manifold can extract all information present in a .prj file; however, users are cautioned that .prj is only a partial standard and should not be relied upon to give the same level of complete projection information that is available in more modern GIS formats.

Select All - Check all the boxes in the fields pane.
Select None - Uncheck all the boxes in the fields pane.
Select Inverse - Check all unchecked boxes and uncheck all checked boxes in the fields pane.

Fields
Check fields to be imported.

Import measure
Creates new drawing containing points, each point of which provides a user-defined measure at that location.

Import Z
Create an additional drawing with points that provide a Z coordinate if Z information is present.

Several formats in common use allow objects such as lines to be defined not just by X,Y coordinates but also by Z coordinates at each coordinate location that defines the line. Checking the Import Z box causes an additional drawing to be created filled with points, each of which has a Z coordinate taken from the accessory Z values in the coordinate sequences that define lines or other objects. This drawing can then be used to create a surface.

Shapefiles and Projections

The experienced GIS operator will use shapefiles only with unprojected data. Unfortunately, many GIS users have been tempted into using shapefiles to save data in projected form.

Shapefiles were not originally designed to save projection information. Recent extensions to ArcView allow the partial saving of projection information in a supplemental .prj file that accompanies the other shapefile files. However, because the notion of a supplemental .prj file has been around only in recent years, most shapefiles that have been published do not use this extension.

A further problem is that some applications use shapefile writing libraries published to Internet that contain errors in how .prj files are written. Although Manifold will read .prj files when they accompany shapefiles and Manifold uses a variety of strategies to deal with invalid .prj data or other flaws, it is still best not to rely upon .prj files.

Shapefile format is therefore best used to interchange data from unprojected drawings. When such shape files are imported into Manifold they will appear in Latitude / Longitude projection without any difficulty. When importing unprojected data from shapefiles, the only variable of concern is which datum was used. Normally, the description of the datum used will be found in whatever descriptive commentary accompanies the drawing.

When importing an unprojected shapefile keep on the lookout for any information that tells you what datum was used for that shapefile. Importing an unprojected shapefile will by default import the shapefile using Latitude / Longitude projection using the WGS 84 centroid. Such imports will be acceptable for many GIS purposes; however, for full accuracy one should find out what centroid (that is, datum) was used for the shapefile being imported and use the Edit - Assign Projection dialog to specify it for the imported drawing.

Be Wary of Projected Shapefiles

Serious difficulties can arise when people project drawings and then save the resulting projected data as a shapefile. In such cases there is no guarantee that the author of the shapefile will warn us that the shapefile
contains projected data and cannot be interpreted as a Latitude / Longitude map. Worse yet, there is no information contained within the shapefiles that specifies the projection and parameters used. If these are not provided within some readme.txt file or (in some cases) an accompanying .prj file or other accompanying documentation, the shapefiles lose their geographic context.

The best first step is to import a shapefile using default settings on the importer. Open the drawing and see what coordinates are reported in the status bar when the mouse passes over the drawing. If they are reasonable latitude and longitude values in the expected range, the shapefile was an unprojected data set using degrees and has been imported correctly.

Since shapefiles cannot save datum information it is always wise to take a moment to find out what datum was used to save the data to the shape file and to select it as the correct datum for this drawing in the Edit - Assign Projection dialog. Quite often the authors of unprojected shapefiles will include documentation with the shapefiles that will inform us what datum was used. Regrettably, not all authors are so considerate.

The case becomes more difficult if projected shapefiles are in play. Projected shapefiles on import may result in a drawing that provides the expected visual appearance when it is opened. However, if the latitude and longitude values in the status bar are nonsensical numbers such as 48593 degrees the shapefile was a projected file. The task now becomes one of finding out what projection and projection parameters were used. These can then be manually specified using the Edit - Assign Projection dialog.

Failing to import a projected drawing from .shp format correctly will lead to unpredictable effects when the drawing is used in a map. If we import a projected drawing from .shp we must tell Manifold the correct projection to use to interpret that data since the format does not save this important information. This is so important that Manifold will nag us to verify the projection of new components to make sure we have reviewed the projection that has been assigned and have confirmed it is the correct projection.

Importing a Projected Shapefile

We import a projected shapefile by first importing it in the usual way and then using the Assign Projection dialog to assign the correct projection.

To import a projected drawing from shapefiles:

1. Import the drawing using File - Import - Drawing. Use default settings.
2. Open the drawing.
3. Use the Edit - Assign Projection dialog to specify the projection information that should be used.

The Edit - Assign Projection dialog allows us to manually provide the correct projection information for the coordinates properties. Manifold will then be able to make sense of the data just imported. If the projected shapefile included a .prj file there may be partial projection information already in the Current Projection dialog. If so and if the information is accurate, that reduces the amount of information we need manually supply.

Do not confuse the use of a chosen projection view in a map window (via Edit - Assign Projection) with the use of Edit - Assign Projection in a drawing window to specify projection information missing from legacy formats. If a projected drawing is imported from shapefiles and we fail to tell Manifold the correct projection to use, that drawing has not yet been correctly imported. Subsequent use of the drawing in maps may cause bizarre effects or lengthy delays as the system attempts to compute a new projection for the map based on fundamentally inaccurate coordinates.

Example
We've imported a shapefile showing the United States into a drawing. When opened, the drawing appears visually as expected.

| Latitude / Longitude | 20834°1.726' E 98451°42.59... |

However, we notice that the status bar reports impossible latitude and longitude values when the mouse passes over the drawing. This tells us that the shapefile contained projected data. We hunt around in the documentation that accompanied the shape file and read the following in a text file:

**The US map is in Lambert Conformal Conic centered on -100, 40 with standard parallels at 33 and 45.**

We are immediately grateful that the author of the map took a moment to tell us this necessary information, albeit in a minimal way. To put this information to work we open the drawing and open the Edit - Assign Projection dialog for the drawing.
This dialog reports the settings Manifold is using to interpret the numbers in the drawing. We can change it to tell Manifold to use a different interpretation. Although this dialog is the same as those used in the Projection system, it does not change the actual data in the drawing. It simply tells Manifold how to interpret what is in the drawing.

In the present instance, the dialog shows that Manifold thinks the data in the shapefile is to be interpreted as Latitude / Longitude data in degrees. We need to change that.
We begin by choosing Lambert Conformal Conic as the projection from the hierarchy of projections available. We then double-click into each of the values to change it to those specified by the author of the map. 

"-100" is obviously a longitude since latitudes range between plus or minus 90. This tells us the author centered the map on longitude -100 and latitude 40. We therefore use 40 for the Center Latitude and -100 for the Center Longitude.

"Parallel" is a synonym for "latitude." We can guess that the 33 and 45 numbers are intended for use as the 1st Standard Latitude and 2nd Standard Latitude respectively. The illustration above shows the dialog as we enter 33 for the 1st Standard Latitude. Press OK to apply the values.
After applying the new coordinate properties there is no change in the appearance of the drawing.

| Lambert Conformal Conic | 98°17.498' W 33°36.207' N |

There is a change in the values reported by the status bar. Now, when the mouse passes over the United States the values reported are the expected latitude and longitude values for this part of the world.

**Comments**

Suppose the author of a projected shapefile did not see fit to provide documentation on what projection settings were used? This is bad news. Without information on projection and parameters the shapefile loses much of its value in a geographic context. We may be able to salvage it by georegistering it if it includes features that can be matched to a known good drawing. Georegistration can be a tedious process, though, so one is forgiven for cursing the author of a shapefile who chose to publish projected data in a shapefile without also publishing the data necessary to use it.

Why isn't Manifold able to know what projection is to be used for a shapefile? That's a limit of shapefile format, not Manifold. Modern formats save projection information. Shape format does not.

**Database Drivers and Shapefiles**

The shapefile standard calls for use of `.dbf` files to save database information within shapefiles. Unfortunately, there is no common standard for `.dbf` in use. `.dbf` originally was the dBase II database format used in early PCs. It lives on as a living fossil database format in modern times. However, since different companies have "extended" `.dbf` in various ways the `.dbf` files accompanying one's shapefiles may be inconsistent from system to system.

Original dBase II `.dbf` format is limited to no more than eight characters for field names beginning with an alphabetic character. `.dbf` file names are also limited to eight ordinary alphabetic characters plus a three character extension. Various recent versions of `.dbf` format (such as, for example, `.dbf` format as used in Microsoft's Visual FoxPro DBMS) may extend the number of characters possible both for field names and for file names.

Note that some ESRI products will create and read shapefiles with unusual `.dbf` file names, even those with spaces in them, depending on the operating system or ESRI product involved. However, in Windows it is a massive violation of `.dbf` spec to use `.dbf` file names with spaces in them for `.dbf` files. Manifold normally uses only the Microsoft standard drivers that are installed in your Windows system; however, shapefile import is an exception because ESRI products are willing to violate the standards set by Microsoft `.dbf` drivers. To deal with such ESRI departures from Microsoft standards Manifold is equipped with a special Manifold `.dbf` driver that is used only when importing shapefiles.

When used to import shapefiles, the Manifold `.dbf` driver is willing to accept nearly any naming of fields and files that are used in the `.dbf` portion of the shapefile. When importing tables from `.dbf` files in all cases other than the import of shapefiles, Manifold always uses the host `.dbf` drivers resident on your Windows system.

**Regional Settings**

When importing data from text file formats like `.csv` or from `.dbf` or shapefiles the Regional Options in the system have to match settings within the file that is being imported. This is because such simple formats are too stupid to understand that in different countries people use different symbols to denote decimal points (that is, either a dot character or a comma). To change Regional Options, do the following:

- Go to the Control Panel and open the Regional Options applet.
- Go to the Numbers page,
- Ensure that the Decimal Symbol is set to ‘.’ (dot),
- Ensure that the List Separator Symbol is set to ‘,’ (comma),
- Press Apply to apply changes if there were any,
- Import the `.csv` file or shapefile desired.
- After import, restore the original settings of Decimal Symbol and List Separator Symbol if you like.
Another option is to invoke the Data Sources applet (hidden within Administrative Tools folder of the Control Panel if you're running Win2K) and create a file DSN for the .csv file you want to import. After creating the DSN configure it to use the comma character as a data separator.

**Shapefiles and PRJ Files**

The use of .prj files together with shapefiles to specify projections is a recent development that, unfortunately, has come into play without any definitive standards being published to precisely specify how the .prj should code its contents. Although the use of .prj files works most of the time, it is something of a haphazard system. If a shapefile is accompanied by a .prj file Manifold will analyze the .prj to try to determine what projection and projection parameters are intended for the shapefile.

Manifold recognizes over 490 projection types used in .prj files and so will be able to extract the projection information in almost all cases, reading more different types of .prj than almost all other software packages that can read shapefiles. However, because .prj files do not use standardized nomenclature for projections it is possible that in some cases it will not be possible to read the desired projection even if the shapefile is accompanied by a .prj file.

Another problem is that some third party applications write flawed .prj files. Manifold contains special code to identify and work around the most common .prj flaws. Manifold will also take reasonable steps to deal with partial or unusual .prj situations. For example, when encountering a .prj file that includes datum information (GEOGCS) but no projection information (PROJCS) Manifold will extract the datum but use the latitude / longitude coordinate system.

Despite these special Manifold features to deal with the idiocy occasionally found in .prj files, it is always possible you may encounter a .prj file that Manifold cannot import in a way that captures the coordinate system supposedly in use. manifold.net is always on the lookout for .prj files containing previously-unknown standards to add to our "zoo" of strange examples. If you encounter a shapefile using a .prj that does not import into Manifold correctly, please contact tech support for instructions on how to FTP it to manifold.net for examination.

**See Also**

Projections and Legacy Formats for a rant on the hassles involved in using some legacy formats.

Import a Shapefile for a step by step example of importing a projected shapefile and setting the correct datum.
Import and Export

Import Drawing - TAB
MapInfo TAB format may be imported by Manifold in cases where the .tab file uses projections also available within Manifold (almost all).

**Select All** - Check all the boxes in the fields pane.

**Select None** - Uncheck all the boxes in the fields pane.

**Select Inverse** - Check all unchecked boxes and uncheck all checked boxes in the fields pane.

**Fields** Check fields to be imported.

**Import formatting** Import formatting for colors and styles used for areas, area borders, lines and points when checked. On by default. Imported formatting will be implemented using thematic formatting if the drawing includes two or more objects with different formatting values.

**Register image** Read image control points embedded by MapInfo. When turned on, this option embeds control points in the image and also creates a drawing with both point objects and control points at the designated locations.

**Formatting**

Tab files can contain color information as well as style parameters for objects. Manifold will read foreground color, background color and style information for objects on import, including formatting for area borders. If no color or size information is present in the file, Manifold will use default format settings.

Because sizes are not the same in MapInfo and Manifold, the .tab importer will attempt match sizes as best as possible.

**Data Types**

On import of a MapInfo .mif or .tab file, Manifold will translate MapInfo logical data type columns into **Boolean** data type. Exporting a Manifold drawing to .mif will export **Boolean** columns into MapInfo logical data type columns.

**Datum Naming**

Formats such as MapInfo .mif and .tab do not actually store the names of datums within the projection information stored in the file; instead, the numeric parameters for the datum are saved. When Manifold scans such a file the system will examine these numeric parameters to find a named datum that is a match and will use that datum's name to identify the datum in use.

In some cases, more than one datum may use the same numeric parameters. For example, there are many national datums that are mathematically identical to some generically named datums. When several datums match the given numeric parameters, Manifold will choose a neutrally named datum in preference to a name that uses a specific country name.

**Important Note when Using 64-bit Manifold Editions**

Due to a lack of required Microsoft facilities in 64-bit Windows systems, Manifold in 64-bit mode cannot export, import, export or link to DB, HTML, MDB, XLS or WKx format files. This includes no access to the **MDB** parts of Manifold MFD and MapInfo TAB imports. The workaround for importing or exporting such files is to launch Manifold in 32-bit mode by using the Manifold System (32-bit) shortcut, perform the export from or import into a .map project file and then re-launch Manifold in 64-bit mode using the Manifold System (64-bit) shortcut.

Linking is more complex: the data must be in some format usable in 64-bit mode within 64-bit Windows systems, such as a SQL Server database. Alternatively, the data can be kept within a Manifold .map project file and linked using the Manifold ODBC driver.
Tech Tips

Depending on MapInfo operation, .tab files containing recently deleted objects may not be imported properly. In such cases, the .tab file may be imported with some or nearly all fields missing. To avoid this situation, before saving the file from MapInfo choose Table - Maintenance - Pack Table to physically delete all elements previously marked as deleted.

Manifold is able to import cases of .tab files that reference shapefiles (.shp), in which case Manifold will automatically import the reference .shp file.

When importing from .tab format Manifold will automatically compare field names given in .tab and .dat files and will choose the longest field name in each case if there is a difference between the .tab and .dat files.
**Import Drawing - TAIF**

TeleAtlas Import Format (TAIF) is normally used for TeleAtlas StreetNet files. Files are normally kept together in a separate directory. Choose any of the files in the directory to start the import process.

**Subtype**  
Choose the **TAIF** subtype: **level 1**, **level 2** or **StreetNet**.

- **Select All** - Check all the boxes in the modules pane.
- **Select None** - Uncheck all the boxes in the modules pane.
- **Select Inverse** - Check all unchecked boxes and uncheck all checked boxes in the modules pane.

**Modules**  
Check modules to be imported. Each module will import as a separate drawing. A map will be created using all the drawings.

**See Also**

Import Drawing - GDF for imports of TeleAtlas MultiNet data sets using **GDF** format.
Import Drawing - MapBase for imports of TeleAtlas (formerly ETAK) MapBase format.
Import Drawing - TIGER/Line

TIGER/Line is a very comprehensive data set created by the U.S. Bureau of the Census. It is issued every year or two to support the Census Bureau's operations. Recent editions have appeared every year or so. TIGER/Line 2000 has appeared in two forms, a "redistricting" version and (in 2001) a full 2000 version that includes Zip Code Tabulation Areas (ZCTAs). ZCTAs provide standardized area representations that approximate the locations of zip code linear features.

Manifold's TIGER/Line importer will read all TIGER/Line versions as of the date of publication of this Manifold release, up to and including TIGER/Line 2006 SE (Second Edition).

Select All - Check all fields. All data fields will be imported with this drawing.

Select None - Uncheck all fields. No data fields will be imported. Tech tip: Use this to uncheck all fields before checking those desired.

Select Inverse - Uncheck all previously checked fields and check all previously unchecked fields. This is a fast way to check only a few columns out of many: uncheck those desired and then push Select Inverse.

Fields pane Check the data attribute fields to be imported.

Compose areas Check to create area objects. Not checking this box will create only lines and points.

TIGER/Line files can import with hundreds of data fields. It is a good idea to become familiar with the TIGER/Line documentation and to import only those fields that are necessary.

How to import a TIGER/Line file:

1. Decompress the file using WinZip or other utility. This creates numerous files ending in various .RTx extensions.
2. In Manifold, use File - Import - Drawing using Files of type "TIGER/Line Files (*.bw, *.rt*)" to open the file ending in the .RT1 extension.
3. Import the fields of interest. Important fields for most users are the CFCC field that specifies what type of line most lines are and the FENAME (Feature Name) and FETYPE (Feature Type) fields that give the names of roads and other features. Uncheck the Compose areas box unless you are an experienced TIGER/Line user.
4. The result is usually three drawings: a "Chains" drawing that contains lines, a "Polygon Points" drawing that contains points for the centroids for what would be all closed areas if the Compose areas box would have been checked and a "Landmark Features" drawing that contains points showing the locations of landmarks of interest to the Census Bureau.

For base maps, most users will use only the Chains drawing and will delete the other two drawings. The Chains drawing contains roads, water features and other lines in a single drawing. The different objects may be separated into different drawings of roads, water features, etc., by selecting using the CFCC field as follows:

<table>
<thead>
<tr>
<th>CFCC Starts with</th>
<th>Objects</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Roads</td>
</tr>
<tr>
<td>B</td>
<td>Railroads</td>
</tr>
<tr>
<td>C</td>
<td>Other Transportation</td>
</tr>
<tr>
<td>H</td>
<td>Water Features</td>
</tr>
</tbody>
</table>

To select all roads, use the Query Toolbar to select all records for the CFCC field Starting with A. Edit - Cut the records out of the "Chains" drawing and Paste them into a new drawing called Roads. Likewise, after selecting all records for the CFCC field Starting with H you can Edit - Cut the water lines out of the original "Chains" drawing and Paste them into a new drawing called Water.
TIGER/Line Files

The TIGER/Line files are extracts of selected geographic and cartographic information from the Census Bureau's TIGER (Topologically Integrated Geographic Encoding and Referencing) database. There is one archived (zipped) TIGER/Line file for each county or county equivalent. The file names consist of "TGR" + the 2-digit state FIPS (Federal Information Processing Standards) code + the 3-digit county FIPS code. For example, "TGR01031.ZIP" is the TIGER/Line file set for Coffee County, Alabama.

Unzipping a TIGER/Line .zip file creates a set of 17 files, one per TIGER/Line "record type" that collectively contain the drawings and tables for that TIGER/Line county. For example, the Redistricting Census 2000 TIGER/Line files consist of 17 record types that collectively contain geographic information (attributes) such as address ranges and ZIP Codes and their Add-On codes for streets, names, feature classification codes, codes for legal and statistical entities, latitude/longitude coordinates of linear and point features, landmark features, area landmarks, key geographic locations, and area and polygon boundaries.

Some counties or statistically equivalent entities do not require all of the 17 record types and therefore have less than 17 files. If the types of data contained in Record Types 4, 6, 7, 8, 9, and Z are not appropriate for a given county or statistically equivalent entity, then the U.S. Census Bureau does not include files for those record types.

The file for each county (or statistically equivalent entity) is identified by the state and county FIPS code after the "tgr" in the file name (for example, tgr42107.rt1). The suffixes used for the record type files have been changed to make it easier to identify each record type file (when working with uncompressed versions of the county files). The suffix consistently is .rtn where n is the record type.

For detailed information on TIGER/Line (including which record types correspond to which features and fields), visit the Census Bureau's site at www.census.gov and drill down into their TIGER pages. Download the TIGER/Line documentation in .pdf form, currently running over 300 pages. The TIGER/Line data dictionary in Chapter 6 of the documentation contains a complete list of all the fields in the 17 record types. Other chapters cross-list the fields by feature attribute and geographic entity type.

ZIP Code Fields

ZIPL, ZIPR and similar fields (providing ZIP codes in TIGER/Line) are imported as fixed-length ANSI text fields. ZIP codes cannot be imported as integers because leading zeros, as in "02138", are significant in ZIP codes.

Notes

The manifold.net team would like to thank Bob Heitzman for his excellent tutorial on TIGER/Line usage within Manifold as well as for his continued advice and his generous contribution of TIGER/Line expertise. The tutorial is available from the manifold.net web site's Resources page.
Import Drawing - UDL

Manifold can create a drawing directly from data stored in a geocoded table. This capability is directly analogous to the creation of a linked drawing from a geocoded table except that instead of the drawing being linked to the originating table it is created as an independent drawing within the Manifold project.

A Microsoft Universal Data Link (UDL) file is a way of capturing an OLE DB connection within a file that may be easily "opened" without the need to deal with complex OLE DB dialogs (except once when the UDL is first set up).

Such imports are accomplished by choosing File - Import - Drawing and then choosing Data Sources () in the Files of type box to launch the Data Source dialog. The desired data source can then be configured and used in the Data Source dialog.

Drawings can be imported from geocoded tables using any of the data access methods, such as connection to an OLE DB data source specified by a UDL file, provided by Manifold. Dialogs used are the same as for creating a linked drawing from a geocoded table, except that linked drawings always invoke the Data Source dialog and the result of an import is an imported drawing rather than a linked drawing.

See Also

The Data Source Dialog
Importing and Linking Tables
Linked Drawings
Create a Linked Drawing from a Geocoded Table
Import Drawing - VCT

Idrisi32, a GIS application, uses .vct format to store drawings. Manifold's .vct importer works automatically to import the drawing with no options.

VCT format is used by Idrisi32 to store vector data. The format consists of two files, a .vct file and a .vdc file that are similarly named, for example, isolines.vct and isolines.vdc. Both files are required to get objects as well as data attributes. If either of the required files is missing the data has been damaged and Manifold will not be able to import the file.
Import Drawing - VMAP

VMap data set products use the "vector product format" (vpf) utilized first by the US Defense Mapping Agency and later by its successor, the National Imagery and Mapping Agency (NIMA). They feature a vast number of files organized into libraries. "VMap" is derived from "Vector Smart Map". The Manifold VMAP importer can also handle World Vector Shoreline Plus (WVS+) data as published in "vpf" form.

VMap data set products are produced in several levels of resolution:

- **VMap Level 0** - Details similar to 1:1 million-scale or 1:2 million-scale paper charts.
- **VMap Level 1** - Details similar to 1:250,000-scale paper charts.
- **VMap Level 2** - Details similar to 1:50,000-scale and 1:100,000-scale paper charts.

VMap Level 0 is equivalent to the famous Digital Chart of the World with slightly different attribute coding. Some, but not all, VMap Level 1 data sets are available to the public in the US. VMap Level 2 may or may not be available to the public. Manifold's VMap importers read all three VMap variations.

VMAP data sets are organized as "libraries" that are kept in DHT files. There is no extension to the DHT file, as see above. In the illustration above we open a DHT file on CD number 155 of the VMap Level 1 collection that cover the entire Earth.
Most VMAP1 files have both a "lib" library and a "rference" library. The lib library contains most of the points, lines and areas while the rference library contains place names and other text labels that will be imported into Manifold as labels.

Library
Choose the library within this DHT to be imported.

Select All - Check all the boxes in the coverages pane.

Select None - Uncheck all the boxes in the coverages pane.

Select Inverse - Check all unchecked boxes and uncheck all checked boxes in the coverages pane.

Coverages
A list of layer categories within the library. Most layer categories will import as several drawings.

Import each coverage to separate folder
A good idea to leave checked: will create a new folder in the project into which all of the components from this coverage will be imported. Helps keep the project organized.

Import Fields
Check to import data attributes along with object geometry.

Import Z
Import elevation if provided for an object. Import Z causes an additional drawing to be created filled with points, each of which has a Z coordinate.

Tip: The fastest way to check just one box is to first Select None and then check the box desired. The fastest way to check all boxes except one is to first Select All, uncheck that box, and then choose Select Inverse.

Typical VMap Contents

Data in most VMap data sets are separated into twelve categories that NIMA calls coverages. Although the word "coverage" in legacy GIS usage normally means "layer", in this case the word is used to mean "category of
layers." In VMap each coverage contains data related to a particular subject category, for example "Transportation."

VMap data sets group coverages within "libraries." A reference library containing four reference coverages is also provided with general information to orient the user. The usual arrangement is that a VMap data set is provided on a CD together with a main DHT file that lists all the libraries on that CD. The reference library contains overview cartographic information showing the very broad area (at low resolution) within which the more detailed libraries on the CD are provided.

VMap Level 0 organizes data sets for the entire world onto four CDs. VMap Level 1 organizes worldwide coverage on over 200 CDs. VMap Level 2, of course, uses many hundreds of CDs. The following VMap 1 coverages are typical of all three VMap versions.

**Data Library**

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bnd</td>
<td>Boundaries</td>
</tr>
<tr>
<td>dq</td>
<td>Data Quality</td>
</tr>
<tr>
<td>elev</td>
<td>Elevation</td>
</tr>
<tr>
<td>hydro</td>
<td>Hydrography</td>
</tr>
<tr>
<td>ind</td>
<td>Industry</td>
</tr>
<tr>
<td>phys</td>
<td>Physiography</td>
</tr>
<tr>
<td>pop</td>
<td>Population</td>
</tr>
<tr>
<td>trans</td>
<td>Transportation</td>
</tr>
<tr>
<td>util</td>
<td>Utilities</td>
</tr>
<tr>
<td>veg</td>
<td>Vegetation</td>
</tr>
<tr>
<td>libref</td>
<td>Library Reference</td>
</tr>
<tr>
<td>tilerref</td>
<td>Tile Reference</td>
</tr>
</tbody>
</table>

**Reference Library**

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>libref</td>
<td>Library Reference</td>
</tr>
<tr>
<td>dbref</td>
<td>Database Reference</td>
</tr>
<tr>
<td>polbnd</td>
<td>Political Entities</td>
</tr>
<tr>
<td>placenam</td>
<td>Place Names</td>
</tr>
</tbody>
</table>

**VMap Import Results**

Importing most VMap "coverages" results in the automatic import of numerous drawings. For convenience, these are organized into maps that correspond to the "coverage." In the example above, we have imported the bnd "coverage" and so have created a bnd map that includes the various drawings listed.
The illustration above shows part of the map in a greatly reduced map window (using camouflage green color for the background, of course!).

**VMap Data Fields**

For specific information on the numerous data fields and other data characteristics of VMap data sets, visit NIMA's page at [http://www.nima.mil](http://www.nima.mil) and download the various online publications that describe VMap.

**See Also**

See the Example Import a VMAP Level 1 File
Import Drawing - WKx

Manifold can create a drawing directly from data stored in a geocoded table. This capability is directly analogous to the creation of a linked drawing from a geocoded table except that instead of the drawing being linked to the originating table it is created as an independent drawing within the Manifold project.

Such imports are accomplished by choosing File - Import - Drawing and then choosing Data Sources () in the Files of type box to launch the Data Source dialog. The desired data source can then be configured and used in the Data Source dialog.

Drawings can be imported from geocoded tables using any of the data access methods, such as from a Lotus WKx file, provided by Manifold. Dialogs used are the same as for creating a linked drawing from a geocoded table, except that linked drawings always invoke the Data Source dialog and the result of an import is an imported drawing rather than a linked drawing.

Note that what can visually appear to be a "table" in a spreadsheet file to the eye might not really be organized as a table from the perspective of Microsoft's table-reading, data access software. If Access cannot find a table to read in the Lotus file, Manifold will not be able to either.

Important Note when Using 64-bit Manifold Editions

Due to a lack of required Microsoft facilities in 64-bit Windows systems, Manifold in 64-bit mode cannot export, import, export or link to DB, HTML, MDB, XLS or WKx format files. This includes no access to the MDB parts of Manifold MFD and MapInfo TAB imports. The workaround for importing or exporting such files is to launch Manifold in 32-bit mode by using the Manifold System (32-bit) shortcut, perform the export from or import into a .map project file and then re-launch Manifold in 64-bit mode using the Manifold System (64-bit) shortcut.

Linking is more complex: the data must be in some format usable in 64-bit mode within 64-bit Windows systems, such as a SQL Server database. Alternatively, the data can be kept within a Manifold .map project file and linked using the Manifold ODBC driver.

Although Microsoft is slowly moving forward towards 64-bit operation of Office and the JET components used by Access and Excel, at the present writing (2010) this is such a total kludge that while it is possible to get 64-bit access to Access or Excel it is just easier for most to continue using the 32-bit versions. See comments in the 32-bit and 64-bit Manifold Editions topic.

See Also

The Data Source Dialog
Importing and Linking Tables
Linked Drawings
Create a Linked Drawing from a Geocoded Table
**Import and Export**

**Import Drawing - XLS**

Manifold can create a drawing directly from data stored in a geocoded table. This capability is directly analogous to the creation of a linked drawing from a geocoded table except that instead of the data table being linked to the original spreadsheet inside the XLS file it is created as an independent table within the Manifold project. The drawing (or drawings, since the import can create more than one drawing depending on the import options) is still linked to the data table via a query, so updating the coordinate values in the data table and refreshing the drawing by using the View - Refresh Data command will move the drawing objects. If desired, the drawing can be unlinked from the query and the data table with the Drawing - Unlink command.

Such imports are accomplished by choosing File - Import - Drawing and then choosing Data Sources () in the Files of type box to launch the Data Source dialog. The desired data source can then be configured and used in the Data Source dialog.

Drawings can be imported from geocoded tables using any of the data access methods, such as from an Excel XLS file, provided by Manifold. Dialogs used are the same as for creating a linked drawing from a geocoded table, except that linked drawings always invoke the Data Source dialog and the result of an import is an imported drawing rather than a linked drawing.

Note that what can visually appear to be a "table" in a spreadsheet file to the eye might not really be organized as a table from the perspective of Microsoft's table-reading, data access software. If Access cannot find a table to read in the Excel file, Manifold will not be able to either.

**Important Note when Using 64-bit Manifold Editions**

Due to a lack of required Microsoft facilities in 64-bit Windows systems, Manifold in 64-bit mode cannot export, import, export or link to DB, HTML, MDB, XLS or WKx format files. This includes no access to the MDB parts of Manifold MFD and MapInfo TAB imports. The workaround for importing or exporting such files is to launch Manifold in 32-bit mode by using the Manifold System (32-bit) shortcut, perform the export from or import into a .map project file and then re-launch Manifold in 64-bit mode using the Manifold System (64-bit) shortcut.

Linking is more complex: the data must be in some format usable in 64-bit mode within 64-bit Windows systems, such as a SQL Server database. Alternatively, the data can be kept within a Manifold .map project file and linked using the Manifold ODBC driver.

Although Microsoft is slowly moving forward towards 64-bit operation of Office and the JET components used by Access and Excel, at the present writing (2010) this is such a total kludge that while it is possible to get 64-bit access to Access or Excel it is just easier for most to continue using the 32-bit versions. See comments in the 32-bit and 64-bit Manifold Editions topic.

**See Also**

- The Data Source Dialog
- Importing and Linking Tables
- Linked Drawings
- Create a Linked Drawing from a Geocoded Table

**Import Image**

**Importing Images**

Import images into Manifold using the File - Import - Image dialog. In addition, images stored within databases using spatial DBMS technology can be imported by connecting to the database using Database Console, clicking on the image to highlight it and then clicking the Import button in the Database Console toolbar.

When using the File - Import - Image dialog the Files of type choice acts as a filter for what is displayed in the browse pane in the Import dialog. Simple file types are imported directly from the Import dialog by choosing the desired type in the Files of type box. More complex imports from data sources such as databases are imported by choosing Data Sources () in the Files of type box and then using the Data Source dialog.

Formats used to save drawings vary widely in their capabilities. Some formats save projection information and some, such as .dxf or many types of .shp files, do not. When a drawing format contains projection information, Manifold will automatically use that projection information.
Formats that do not preserve projection information (the majority) will require manual intervention to place and scale the image correctly within a geographic context. Most often the image will be placed in a geographic context through georegistration.

**Note:** USGS DOQ files as provided on CDs are normally not georegistered. Most often these are .jpg format files created from the original DOQ format files and provided on CD without the .jpg extension.

See the Projections and Images topic for an introduction to projection issues involving images.

**Importing Satellite Image Formats**

Many formats used for satellite images are located under File - Import - Surface because such formats are often used to present non-visual data as raster data sets. See the Importing Surfaces topic.

For example, HDF, HDF EOS and HDF SeaWiFS formats are found in the File - Import - Surface dialogs. If you don't see a desired format listed in the image importers, check under surfaces.

When importing HDF files, use HDF SeaWiFS for SeaWiFS data in .hdf format and use HDF EOS for Earth Observation Satellite data in .hdf format. While the ordinary HDF converter will also import some data from HDF SeaWiFS files it will ignore multidimensional data chunks (called Scientific Data Sets or SDSs) that are used to contain pixels. Note also there is a CEOS SeaWiFS importer for data in that form.

Some file types, such as ENVI, ERDAS and ERS (ECW) files, can contain either surface or image data, depending on how the data is intended to be interpreted. The ENVI, ERDAS and ERS (ECW) importers occur both in the File - Import - Surface and File - Import - Image menus and have a subtype option to allow specification of image or surface upon import.

**Importing Georegistered Images**

The preferred format for importing georegistered images is GeoTIFF format. This standard is supported by over 150 companies and provides for reliable import of images with correct georegistration. Some other formats, such as ECW can also save projection information.

**Use of World Files**

An obsolete way of providing partial projection information for images is the use of "world" files. World files do not contain the name of the projection used. They simply provide certain local projection values and leave it up to the user to manually specify the actual projection in use. A world file is a small text file that accompanies the image file. For example, an image called Europe.jpg that is an ordinary graphics file in JPG format might be accompanied by a small file called Europe.jpegw, which is the world file. The Europe.jpegw file is an ordinary ASCII text file containing a few lines of text. The extension ".jpgw" is used to let GIS software know that it is a world file.

If we opened up the Europe.jpegw world files in Notepad we would see something like the following:

```
4.000000
0.000000
0.000000
-4.000000
574400.000000
4145600.000000
```

The numbers provide information on local offset and local scale for whatever projection is in use. However, the world file does not actually say what projection has been used. When publishing images accompanied by world files the assumption is that the user knows what projection is supposed to be used and is capable of manually specifying it in the GIS software being used.

This is a really stupid system, of course, since the moment images are distributed away from their author and other people who know the secret of what projection was used there is nothing in the world file to tell the rest of normal humanity what the author intended. If you are using images accompanied by world files, it is critically
important to search carefully through the website from which you downloaded the images to find any information that describes what projection was used. If you cannot find a file or web page that specifies the actual projection that was used the world file alone will not help you georegister the image.

Once you know the projection you can georegister images accompanied by world files by adding the local projection parameters they specify to the projection in use. Manifold makes this easier by automatically reading the local projection information from a world file that accompanies an image file. When loading images from TIF, BMP, GIF, JPG, PCX, PNG and TGA formats, Manifold scans the directory to see if there is a similarly named file having one of the two common world file extensions. If so, Manifold opens the world file and loads the local projection information from the world file.

For TIF, BMP, GIF, JPG, PCX, PNG and TGA formats the world file extensions are usually in the form .tifw, .bmpw, .gifw, .jpw, .pcxw, .pngw and .tgaw respectively; however, some world files may have extensions in the form .tfw, .bpw, .gfw, .jgw, .pxw, .pgw and .taw respectively. Manifold will automatically read world files having either the ".tifw" or ".tfw" style of filename extension.

Once the image has been imported, it must be opened and the Edit - Assign Projection dialog must be used to specify the correct projection. See the Download and Mosaic Terraserver Images topic for an example of this process using world files and images.

**Typical Image Formats**

Manifold can import from many different image formats. Some are highly specialized formats that provide much data organized in a complex way. Precise description of certain specialty formats is beyond the scope of this document. In such cases, please refer to the originating agency or company for detailed documentation on their format.

Interested in support for a new format not listed below? See the Contacting manifold.net topic for information on suggesting a new format.

Searching Internet with a good search engine will find numerous references to format documentation as well. For example, searching for "PNG" and "Portable Network Graphics format" on Google turns up the PNG home page as the first hit.

- **ADRG** NIMA Arc Digitized Raster Graphics.
- **BMP** Microsoft .bmp graphics format. Imported as a partially georegistered image if accompanied with a .bpw or .bmpw "world" file. Manifold can import .bmp images that are RLE compressed.
- **CADRG / CIB** NIMA Compressed Arc Digitized Raster Graphics (CADRG) and Controlled Image Base (CIB) images. Get free downloads of Operational Navigation Charts (ONC), Tactical Pilotage Charts (TPC) and Joint Operations Graphic - Air (JOG-A). The importer can import individual tiles as well as a composite image defined by an .toc file. Since composite images can be huge, importing individual tiles is a great space and time saver when the entire image is not required.
- **DNG** An Adobe variation of TIFF used in digital photography.
- **DOQ** USGS Digital Ortho Quad images. True "DOQ" files occur with a .doq or .coq extension. "DOQ" files are often provided in the form of .jpg compressed jpeg format images. Use the .jpg image importer for those
- **ECW / JPEG2000** Compressed images in either ECW or JPEG2000 compressed format. Any size ECW or JPEG2000 image may be imported.
- **EMF** Microsoft Windows Extended Meta File (.emf) and Windows Meta File (.wmf) format. Images imported from will have their dimensions adjusted to not exceed 4096 pixels in X or Y dimension. This limitation avoids possible problems in the Microsoft GDI driver. Images imported
will use white background color.

**ENVI IMG**  Image/surface format from ENVI, "The Environment for Visualizing Images". Import uncompressed ENVI .img files. (Manifold does not support compressed ENVI image import.) In addition to the usual image types, the importer will import 12 bit images from .img files.

**ERDAS GIS**  ERDAS files in .gis or .lan formats. Imported using the File - Import - Surface dialog.

**ERDAS IMG**  ERDAS files in .img format. Works for IMAGINE format .img files.

**ERS**  Earth Resources Mapping (ERMapper) images or surfaces in .ers files. .ers files saved with projection information will be correctly georegistered on import if the projection used is one supported by Manifold. Since this format can be used for either images or surfaces, when importing choose the subtype (either surface or image) desired once the importer launches.

**GIF**  Common .gif graphics files. Imported as a partially georegistered image if accompanied with a .gfw or .gifw "world" file.

**HDF**  A graphics format often used in remote sensing.

**JPEG**  Standard .jpg and .jpeg format, all types. Imported as a partially georegistered image if accompanied with a .jgw or .jpgw "world" file. Imports EXIF tags as well.

**NITF**  National Imagery Transmission Format - a format used mostly by military users in the US and UK for images. Also used for some satellite images generated by high-resolution commercial satellites. Often has .ntf extension. Any extended data (such as text) in an NITF file will be imported into a comments component.

**PCX**  PC Paintbrush format by ZSoft. Imported as a partially georegistered image if accompanied with a .pxw or .pcxw "world" file.

**PNG**  Portable Network Graphics (PNG) format, pronounced "ping." Imported as a partially georegistered image if accompanied with a .pgw or .pngw "world" file. Billed on the current PNG home page as "A Turbo-Study Image Format with Lossless Compression." The original acronym is said to mean "PNG's Not GIF." This is a superb format that can be used to replace GIF everywhere (even on web pages) with better compression and higher quality images.

**PPM**  Portable PixMap (PPM) format. Intended as a "lowest common denominator" color image file format. Used in UNIX systems.

**SGI**  Silicon Graphics workstation graphics format.

**SID**  "MrSID" images using LizardTech proprietary format. Requires prior installation of the free LizardTech MrSidDecode.exe utility. See the Import Image - SID, MrSID help topic for details.

**SPOT**  SPOT satellite images, often provided in 10-meter resolution.

**SRTM**  NASA Shuttle Radar Topography Mission data using files in one or more of the following file formats:

- .hgt - Height. Files import as Int16 surfaces.
- .err - Error data. Files import as Int16 surfaces named "…Error".
Import and Export

.import - Incidence angle data. Files import as Int16 surfaces named "...Incidence".

.imag - Image. Files import as grayscale images named "...Image".

.pol - Polarization angle. Files import as palette images named "...Polarization".

SUN Sun workstation graphics format.

TGA AT&T Truevision Targa format. Imported as a partially georegistered image if accompanied with a .taw or .tgaw "world" file.

TIFF Tagged image file format, using .tif and .tiff extensions. Manifold's .tif importer reads all types of .tif files including GeoTIFF georegistered images and .tif images partially georegistered using a .tfw or .tifw "world" file.

(Various DBMS formats) Manifold can import images from information stored in tables or created by queries. Imported using the Data Source dialog. See the discussion below.

DB2 Data Sources Connect to IBM DB2 equipped with the DB2 Spatial Extender using IBM's native spatial connection technology. A spatial DBMS connection accessed through the Data Source dialog. This option requires Enterprise Edition or above.

Oracle Data Sources Connect to an Oracle data sources using the Oracle Call Interface (OCI), the native Oracle interface for exchanging data. Exchanging images with Oracle data sources via OCI automatically maps images into Oracle GeoRaster form (if GeoRasters are supported by the Oracle product in use). A spatial DBMS connection accessed through the Data Source dialog. This option requires Enterprise Edition or above.

PostgreSQL Data Sources Connect to PostgreSQL equipped with the PostGIS spatial extender using native PostgreSQL connection technology. Connections made using this data source will have any password required masked so that later usage will not inadvertently expose the password. A spatial DBMS connection accessed through the Data Source dialog. This option requires Enterprise Edition or above.

SQL Server Data Sources Connect to Microsoft SQL Server 2008 spatial DBMS facilities using native SQL Server 2008 spatial connection technology. A spatial DBMS connection accessed through the Data Source dialog. This option requires Enterprise Edition or above.

Important: Using a vendor's native spatial DBMS connection technology (like those above) provides better performance and allows using features, such as built-in spatial DBMS facilities, not exposed through generic database interfaces such as ODBC, OLE DB and ADO .NET. At the present writing, only Oracle Spatial provides a built-in GeoRaster type. Other spatial DBMS products are supported using Manifold-managed image and surface storage.

Note: because the Database Console provides a more convenient user interface for browsing databases, we will almost always use the Database Console to import or link components from a database instead of using the File - Import or File - Link menu commands.

XML Files Created upon Export and Used on Import
Not all formats to which Manifold can export have the ability to correctly save projection information. As a safety measure to ensure that projection information is never lost, Manifold always writes an accompanying .xml file when exporting drawings, images, labels, maps or surface components to a file. The accompanying .xml file contains coordinate system information and some other metadata. The XML file is created for all formats, even those that correctly save coordinate system information.

When Manifold imports drawings, images, labels or surfaces, the system will check for an accompanying .xml file that might have been written by Manifold. If such an .xml file exists, Manifold will read it and use the information it contains to load a correct coordinate system.

See the XML Accessory File Format topic for details on the internal organization of this file.

Imported or Linked Images from Tables or Queries

Manifold can import images or link images from information stored in tables or provided by queries by using the Data Source dialog. The tables and queries used to create linked images may be inside the Manifold project or they may be in external databases.

Linked images may also be created in a two step process, where the virtual table for an existing image is used in a query to manipulate that image or to fetch part of the image based upon desired criteria. A linked image can then be created from that query. This technique is often used within Manifold projects that will be used in an IMS website.

Linked images may also be created from tables or queries stored in external databases. For example, we could use a query to select all columns for all pixels in an image's virtual table and then export that query as a table into an .mdb file or to some other database storage.

When an image is linked from a table or query, each record in the table or query produces a pixel. The coordinates of the pixel are taken from the X and Y columns (cast to integer) specified in the Link dialog. The color of the pixel is taken from either the Color column or the channel columns.

If you have Manifold System Enterprise Edition or greater you can also import images or link images from Oracle data sources that use Oracle's Spatial or Locator technologies. Importing images from Oracle data sources is virtually identical to linking them from Oracle data sources. See the Linked Images from Oracle Servers topic.

Importing or linking images from databases is normally accomplished using the Database Console to connect to the DBMS, and then the Import or Link Database Console toolbar button is used to import or link the desired image from the database.

Importing Images from Image Servers

In addition to the straight File - Import - Image importers within Manifold we can also import images by first creating a linked image that is dynamically fetched from an image server and then unlinking or downloading that linked image to create a local image. This allows us to import images covering a desired area from image servers. See the Linked Images topic to learn more about linked images and image servers.

Important Note when Using 64-bit Manifold Editions

Due to a lack of required Microsoft facilities in 64-bit Windows systems, Manifold in 64-bit mode cannot export, import, export or link to DB, HTML, MDB, XLS or WKx format files. This includes no access to the MDB parts of Manifold MFD and MapInfo TAB imports. The workaround for importing or exporting such files is to launch Manifold in 32-bit mode by using the Manifold System (32-bit) shortcut, perform the export from or import into a .map project file and then re-launch Manifold in 64-bit mode using the Manifold System (64-bit) shortcut. Linking is more complex: the data must be in some format usable in 64-bit mode within 64-bit Windows systems, such as a SQL Server database. Alternatively, the data can be kept within a Manifold .map project file and linked using the Manifold ODBC driver.

See Also

Projections and Images

Images and subsequent topics.
Historical Notes

Some graphics formats popular with the graphics communities arise from formerly elite machines used before PCs redefined the meaning of "elite." In case the companies involved are gone or forgotten by the time this is read, a few notes for younger readers:

SGI was an elite graphics company in Mountain View that ended up as a sort-of UNIX workstation vendor famous for three things: 1) Inventing OpenGL, 2) Having a very cool building across the street from the Century 16 cineplex in Mountain View that now, appropriately enough, serves as a museum of computer technology, and 3) Last, but not least, failing to keep up with the price/performance competition of Wintel clones running cool new graphics engines from companies like NVIDIA.

SUN once was an elite UNIX workstation company in Palo Alto before clones banished it to a netherworld existence as a server appliance company. Sun was yet another company that started with a plug-in graphics card, a Multibus card for the "Stanford University Network" (SUN) machines then in use at Stanford. They went on to bigger and better things, including a respectable workstation and server business. Unfortunately, they were run over by the steamroller of high power / low cost clones running ever more applications-rich Windows operating systems, and "workstation" became just another synonym for "PC."

AT&T at one point was the fearsome owner of all telecommunications mojo in the US. They also (we're not making this up) built a "high performance" graphics card called the Targa that plugged into PCs. The AT&T brand lives on as a new identity for one of the "baby Bells" that ended up consuming the parent and is now busy gobbling up cell phone companies.
Import Image - JPG, JPEG

JPEG is a standard for image formatting saved in files that end in a .jpg or .jpeg extension. Manifold recognizes both. For brevity this topic uses only the .jpg extension and should be understood to apply to both .jpg and .jpeg files.

The "JPG" importers in Manifold can import various forms of .jpg or .jpeg files:

- Ordinary .jpg images.
- .jpg images that are accompanied by a jfw "world" file to (hopefully) provide georegistration information.
- .jpg images that contain EXIF tags.

The JPG importer will scan the contents of the .jpg file and automatically select the correct import mechanism.

Ordinary .jpg files are simple graphics files that have no georegistration information. They import as ordinary images and must be georegistered using Manifold’s georegistration tools.

JFW World Files

Sometimes .jpg files intended for geographic usage in a GIS may be accompanied by "world" files in .jfw format. This combination uses two files, a .jpg file and a similarly-named .jfw "world" file to provide some georeferencing information. The .jfw file provided in such cases is not part of the JPEG standard but rather is an auxiliary file invented by GIS types who for whatever reason did not have the ability to use a truly geographically-aware image file format. The use of world files is an incomplete hack that causes much trouble for novices.

When a .jpg is imported Manifold checks for a .jfw file and does its best to import using whatever information may be found in that file. If a .jfw file is not found Manifold will import the .jpg as an ordinary graphics file without any georegistration.

Limitations of JFW files

JFW’s often will not have all of the information required to georegister an image. For example, some JFW files will only hold offsets and scales (which are correctly read by the importer) but these are not enough to register an image to a location and projection on Earth. For this reason, GeoTIFF or other geographically-aware graphics standards are better standards to use for interchange than JFW.

On occasion it is possible to rescue a JPG/JFW that does not contain complete georegistration information. Many JFW files are saved using Latitude / Longitude values for the pixels in the accompanying .jpg image. The actual .jfw file is a plain text file that can be opened with Notepad or some other text editor and inspected to see what it contains. We can sometimes guess by looking at the scales, offsets and dimensions numbers in the JFW that the intended projection system is simple Latitude / Longitude.

If the data is in Latitude / Longitude projection it is fairly easy to import the files and add the necessary projection information by changing the values in the Edit - Assign Projection dialog.

To import a .jpg / .jfw in Latitude / Longitude projection:

1. Import the file using the JPG importer.
2. Open the Edit - Assign Projection dialog. The coordinate system will be reported as the default Orthographic projection with offsets and scales from the JFW correctly read into Local Offsets and Local Scales.
3. Write down the values for Local Offsets and Local Scales.
4. Change the projection to Latitude / Longitude.
5. Modify Local Offsets and Local Scales to the values written down in step 3 above.
6. Check to make sure the coordinate system units combo is set to Degree.
7. Press OK.
Note that if one cannot determine the intended projection of the .jfw by inspecting the file there may be some accompanying documentation that describes the intended use of these files.

**EXIF Tags**

Exchangeable Image File Format (EXIF) is a standard used mostly by digital cameras for adding information to JPEG images produced by those cameras. EXIF tags can contain a very wide variety of information that is inserted into images by camera programmers. Manifold recognizes both EXIF tags as well as EXIF tag directories written by major camera vendors.

If a .jpg image contains EXIF tags, Manifold will write the information from those tags into the Notes pane properties for that image. To see the EXIF information, open the image: with the focus on the opened image the Notes pane will show the EXIF information for that image.

.jpg images containing orientation EXIF tags will be rotated and flipped as necessary to use the orientation specified.

**XMP Tags**

Extensible Metadata Platform (XMP) is an XML based metadata format used to embed a seemingly endless variety of information into images. If a .jpg image contains XMP tags Manifold will read the information into the component notes as is done with EXIF tags.

**Comments**

In addition to use with JPG files, Manifold supports the use of "world" files to import georegistered images from other formats, specifically from BMP, GIF, JPG, PCX, PNG, TIF and TGA files using JFW-like "world" files (with BPW, GFW, JGW, PXW, PGW, TFW and TAW extensions respectively).
Import Image - SID, MrSID

Multiresolution Seamless Image Database (MrSID) is a closed, proprietary format for storing images originally developed at the US Government's Los Alamos National Laboratory in the early and mid 1990's and then exclusively licensed in a somewhat aberrant transaction to a small private company. Despite the general replacement of MrSID by more modern formats such as ECW and JPEG2000 (both of which are true open formats), there are still legacy collections of images that utilize MrSID format.

There are three ways of getting MrSID files into Manifold:

- In Windows, open a command prompt window and use the LizardTech mrsiddecode.exe utility to convert MrSID files into GeoTIFF and then import the GeoTIFF. This utility can be used in a background command prompt batch file so that the long time required can be invested during off hours.
- In Manifold, use the Import - Image dialog for MrSID files. This Manifold capability depends upon having the mrsiddecode.exe utility or equivalent installed on the computer and is equivalent to the above. This is a good idea only for smaller MrSID files since the process of conversion and import takes so long.
- Use a third party utility or script that converts MrSID into ECW or some other modern format. The ECW files can then be instantly opened in Manifold.

Using the Manifold Import - Image dialog, the MrSID files can take nearly forever to import into Manifold because they are first converted by a LizardTech or other vendor's utility into GeoTIFF and then the GeoTIFF file is imported into Manifold. Depending on the speed of the hardware used, the version of windows, the amount of RAM available and other system configuration issues, the conversion and import process takes tens of minutes for small files, hours for medium or large images, many hours for larger images and days for very large images.

Even using the mrsiddecode.exe converter as a standalone program in a command prompt window the conversion process is very slow because the conversion process can go no further than the speed of the converter provided by LizardTech, which is not known for speedy performance. Users should be forewarned that a MrSID image occupying 100MB on hard disk as a .sid format file can expand into tens of gigabytes as a GeoTIFF file and take days to convert. Recent editions of the LizardTech decoder are said to be faster, but users should nonetheless brace themselves for a trial by endurance.

Comments published on various web sites have expressed the opinion that the LizardTech converter has artificially been made slow to discourage users from abandoning MrSID format in favor of more modern alternatives. Whether that is true or not, it is certainly true that there are converters available that have been written by third parties which are said to operate much faster than the LizardTech converter and which use the same command line interface. Manifold can work with any such compatible converter as well.

Whatever the speed of the decoder used, the wisest strategy for dealing with MrSID images is usually to acquire or to create a small Windows command prompt batch file to use the LizardTech or other decoder to convert all MrSID images in background to a more modern format like ECW, running the decoder at night or during the weekend. If this process is accomplished before the images need to be used there will be no inconveniences from delays in importing when using the images in Manifold. See the comments at the end of this topic on how to accomplish this strategy.

Until that strategy is accomplished, the Manifold Import - Image dialog used for MrSID import as discussed in this topic should be used only for relatively small MrSID files when sufficient time is available, such as when taking a break or going to lunch, for the process to be completed.

Requirements for Importing MrSID Files

Manifold can import images from MrSID format if a compatible MrSID decoder has been installed on the same computer. Manifold can work with any MrSID decoder that is compatible with the mrsiddecode.exe decoder distributed by the LizardTech company, the current licensees of MrSID. At this writing, the mrsiddecode.exe decoder may be downloaded at no charge from the www.lizardtech.com website. Drill down through the site to find the download page for this utility.

A possibly better idea than using the LizardTech tools (which run unnecessarily slow) is to use MrSidDecodeFast or MrSidExtract or similar third party tools that run much faster.

Since websites change frequently, if you cannot find the decoder on the LizardTech website, use Google or some other search engine to find the current location for downloads of that utility. For example use MSN search or Google to search for "MrSidDecodeFast.zip" or "MrSidDecodeFast" to locate these utilities.
Since company policies frequently change, if the necessary decoder is not available for free at some time in the future from the LizardTech site, you may use any other compatible MrSID decoder by changing the name of the .exe file to use in the Tools - Options - File Locations dialog to the name of the decoder.exe you use. For example, you could change it to MrSidDecodeFast.exe if you have downloaded that free third party utility.

To prepare your system for MrSID use:

1. Login as Administrator if running Windows 2000, XP or 2003.
2. Visit the www.lizardtech.com site and find and download the mrsiddecode.exe and mrsidinfo.exe utilities. These may be packaged in zip files that may be downloaded and then unzipped to get the .exe files they contain.
3. Place both .exe files somewhere in the PATH defined for executable programs on your Windows system, such as in the C:\WINDOWS or C:\WINNT folder.
4. Verify the programs have been installed and are functional by opening a command prompt window in Windows and doing a mrsiddecode.exe -help command. If it runs OK you know your .exe has been installed correctly.
5. Launch Manifold and in the Tools - Options - File Locations dialog verify that the correct names for the mrsiddecode.exe and mrsidinfo.exe files are entered.

Once the mrsiddecode.exe and mrsidinfo.exe utilities have been installed on your system and their names have been correctly entered into the Tools - Options - File Locations dialog, you may use File - Import - Image to import MrSID format images into Manifold.

To import a MrSID image:

1. Choose File - Import - Image
2. In the Import Image dialog's Files of type box choose SID Files (*.sid)
3. Browse to the file desired and double-click on the file to be imported.

**Import Image Dialog Options**

- **Scale**: Scale reduction factor ranging from x1 (native) (full resolution) to x64 (one sixty-fourth the resolution). Use this when importing large MrSID images to automatically create a smaller, albeit lower resolution, image.
- **Use password**: Check this box to supply a password if the MrSID file requires a password.

MrSID images are very large images and even small MrSID files can take a substantial amount of time (tens of minutes or even longer) to import into Manifold. The process is lengthy because first the decoder utility program is used to decode the MrSID image and then the decoded image must be imported into Manifold. The entire process of using an external utility is hidden from the user with Manifold showing only the standard import and progress bar dialogs.

By default, Manifold will automatically set the Scale factor so the resulting image will contain less than 4 million pixels. For large MrSID images, this will result in an image with substantially less resolution than is possible. To import a full resolution image, use the x1 (native) setting for Scale. **Caution**: large MrSID images can be very slow to import due to the incredibly slow functioning of the LizardTech MrSidDecode utility. As a safety measure, Manifold will raise a confirmation dialog when commanded to import a MrSID image with more than 16 million pixels.

Since MrSID has largely been replaced in modern usage by open formats such as ECW and JPEG2000, Manifold will import MrSID images but it will not export images to MrSID nor utilize MrSID as a native format for linking images as compressed images as is possible with ECW and JPEG2000.

The assumption is that if you are working with Manifold and must deal with an image in MrSID format, your first objective will be to immediately convert the image into a compressed image and to save it out as an ECW or JPEG2000 format image. Once the image has been converted into and saved as a compressed image, loading it and viewing it will be instantaneous and it no longer will be imprisoned in MrSID format.
Because many images are trapped in MrSID format and most GIS users would prefer to use a more open format, some software developers who have licenses to work with ECW or JPEG2000 have created scripts or utilities such as MrSidExtract.exe that can automatically traverse a hard disk and find and convert all MrSID files into ECW files. That is the most efficient way of freeing images trapped in MrSID format since the lengthy process of converting a MrSID image to a more modern format can be accomplished at night or over the weekend.

Although at the present writing the mrsiddecode.exe and mrsidinfo.exe utilities may be freely downloaded from the www.lizardtech.com website, it is not clear exactly what their licensing agreement says about who may or may not use those utilities. If these utilities disappear or if you disagree with their licensing terms, you may be able to find alternative decoders on the web that perform the same function.

Manifold.net has not tested any alternative decoders, but just as there was once only a single MP3 encoder/decoder and now there are many including MrSidDecodeFast. Manifold includes the ability to work with alternative decoders should such emerge in the future so long as they are compatible with the mrsiddecode.exe and mrsidinfo.exe utilities.

Projections and MrSID Files

MrSID format can optionally store projection information for the contents; however, many MrSID files do not contain such information. Such files may or may not be accompanied by .sdw "world" files. Unfortunately, the LizardTech code for decoding MrSID files cannot be relied upon to capture either any projection information within the MrSID file or from the accompanying "world" file, if any. Manifold has facilities to deal with these issues should the need arise.

Although in theory the mrsiddecode.exe decoder created by LizardTech should capture any coordinate system (projection) information available for the MrSID file when converting to GeoTIFF, as a practical matter there is no guarantee that it will do this correctly in all cases, since it might not know the GeoTIFF equivalents for certain coordinate systems that can occur in a MrSID file. It would be unwise to expect the LizardTech code to accurately capture projection information in all cases: instead, the experienced user is always ready to manually assign the desired projection to the imported image.

When relying upon third party decoders, such as MrSidDecodeFast or MrSidExtract there is no guarantee there will be any projection information inside the MrSID or that the third party decoder will be able to do anything with it. In such cases it is wise always to assume the worst and to manually assign the desired projection to the resultant ECW or image.

When manually assigning projections, we can consult any metadata or readme files that accompany the imagery, as well as manually opening and reading any "world" files that might accompany the images. See the discussion on importing shapefiles and in the Importing Images topic for a forewarning of the hassles that may be encountered using "world" files and strategies for dealing with those hassles. Users of MrSID files should be ready to assign a correct projection manually after importing such images using the Manifold facilities provided for assigning projections.

Users of MrSID files should also be forewarned that MrSID compilations that are published by various government bodies historically have had a much higher percentage of erroneous metadata descriptions than any other format known to Manifold Technical Support. Users routinely report problems of misaligned MrSID images that were obtained from some government web site, imported into Manifold, the projection described by the metadata carefully assigned and then when the image was found to be inaccurately registered the problem was ultimately traced to inaccurate metadata published on that web site.

The Strange History of MrSID

According to the Los Alamos National Laboratory website, MrSID technology is based upon two Los Alamos inventions: MCICR and MrSID.

Monte Carlo Image Conversion and Representation (MCICR) technology was developed at Los Alamos by Vance Faber, James White, and Jeffrey Saltzman. MCICR consists of two patented techniques for compressing 24-bit digital images into smaller 8-bit images without loss of resolution.

Multiresolution Seamless Image Database (MrSID) was developed by Jonathan Bradley through the Los Alamos laboratory's Sunrise Project funding. The combination of MCICR with MrSID produced image technology that became known as MrSID.

As described on the Laboratory's web site, "Before licensing Los Alamos' technologies, Paradigm Concepts, Inc. was a small service-based New Mexico company consisting of 4 employees. Since licensing MCICR and MrSID, the company has grown to 55 employees, including Bradley, Faber, and White - three of the original developers."

Also quoted on the Laboratory's web site was the then-President of LizardTech, John R. "Grizz" Deal, stating, "The creation of the MrSID technology could have only happened at the world's premier applied research institution, Los Alamos National Laboratory. Without MrSID, LizardTech would not be in business today."

According to news reports, LizardTech grew considerably after the above was written, and at its peak employed approximately 150 people. However, the company apparently never achieved a solid financial footing and spent considerable effort in an obnoxious legal action against Earth Resource Mapping, the company that created ECW, a competing image compression technology.

LizardTech was apparently trying to win in court what it could not accomplish in a competitive marketplace. The lawsuits failed, in some cases so miserably that courts ruled against LizardTech in summary judgments. Nonetheless, the legal actions consumed large sums in legal fees. A further negative outcome was that LizardTech clouded the status of open standards like ECW and JPEG2000 with potential patent claims, however unfounded, and so caused unnecessary expense within the open source community as LizardTech's claims in court were examined, refuted and ultimately rejected.

Although LizardTech was reported to have raised over $44 million in venture capital money, by 2001 the company was already making massive layoffs, firing about half of its employees, with more layoffs in 2002 as well. It appeared to be well on its way to wasting a staggering amount of venture capital.

By 2003 LizardTech had managed to bum through tens of millions of dollars of venture capital and was forced to fire almost all of its employees and sell the remaining assets to a Japanese company, Celartem Technology Inc. As of this writing, it remains to be seen if Celartem can revive MrSID despite the market preference for open technologies such as ECW and JPEG2000.

Why is MrSID Conversion and Import so Slow?

The incredibly slow functioning of the LizardTech mrsiddecode.exe program when converting MrSID images to an open standard like GeoTIFF appears to be yet another negative result of having to work with a seriously "closed" format like MrSID. Because the read/write technology inside the decoder is also closed, users are kept hostage by whatever inclination LizardTech may or may not have to improve the functioning of the decoder. Some critics have expressed the opinion that LizardTech has provided a deliberately slowed-down conversion utility to make it difficult to convert images from MrSID format to modern, open formats like ECW or JPEG2000.

Considering that LizardTech's historical strategy has been to maintain MrSID as a deeply closed format, it is obvious that LizardTech has every incentive not to try to speed up a decoder that allows people to free themselves from MrSID captivity. If anything, a cynic would point out that LizardTech would want to offer a decoder that works well enough for the company to say that MrSID can be converted to open formats but which works so poorly that as a practical matter few people will be able to use the decoder to free themselves from MrSID.

Contrast that situation to the much more open world of ECW or JPEG2000, where source code is easily available and as a result anyone who wants to contribute to improved performance can do so. As a result, both ECW and JPEG2000 provide at least four times faster performance in “native” compressed format applications and allow conversion to other formats using dramatically faster processes than made available by LizardTech.

The recommended Manifold strategy for dealing with MrSID is to convert MrSID format images into ECW format. Although LizardTech may not provide any tools for that purpose, there are third party tools such as MrSidExtract.exe that can do an automatic conversion of all MrSID files in a given folder or folders into ECW.

Many people have written scripts or programs like MrSidExtract.exe that can search through a folder or even a hard disk full of MrSID files and in background accomplish the above two-step process to convert those images into ECW. If accomplished in background, such as at night or during weekends, the tremendously slow process of getting images out of MrSID and into fast, modern formats like ECW can be done without the interactive pain of sitting and watching a conversion process taking forever. The resulting ECW images can be loaded into Manifold and viewed in seconds.
Such scripts or programs are not a part of Manifold and so are not covered by this documentation. Use a search engine like MSN search or Google or search Manifold forums such as the Georeference forum on the manifold.net home page and the archives of the Manifold-L discussion list to find links to pages that provide such scripts.

**Tech Support**

Although Manifold works with the LizardTech tools and other third party tools, such tools are not a part of Manifold System and are not supported by Manifold Technical Support. As noted above, tools vary greatly in performance, memory requirements, ability to extract projection information and other characteristics. These characteristics are not under the control of manifold.net, nor does manifold.net have any inside ability to either analyze or repair shortcomings in non-Manifold products. If you experience any unexpected results when using a third party tool, contact the source of that tool for assistance.

**See Also**

Tools - Options - File Locations

Compressed Images
Import and Export

Import Image - TIF [Various types]

The "TIF" importers in Manifold can import different types of .tif or .tiff files (*.tiff files are treated the same as *.tif files and vice versa):

- Ordinary tif images that do not have embedded georegistration information.
- DRG's and other tif images that are accompanied by a tfw "world" file to (hopefully) provide georegistration information.
- .tif images written in GeoTIFF standard. This standard embed projection information using "tags" within a single tif file.
- .tif images encoded using 11-bit data.
- .tif images encoded using 16-bit data.
- .tif files containing 16-bit integer LAB data.
- .tif files containing 16-bit integer and 32-bit / 64-bit floating point CMYK data.
- .tif files containing complex (that is, real + imaginary) number values.
- .tif images that contain EXIF tags.

The TIF importer will scan the contents of the .tif file and automatically select the correct import mechanism. Surfaces imported from .tif files with floating-point values will automatically set pixels with invalid or infinite floating-point values invisible.

Ordinary .tif or .tiff files are simple graphics files that have no georegistration information. They import as ordinary images and must be georegistered using Manifold's georegistration tools.

TIF or TIFF format files can occur in a specialized form called GeoTIFF files that save projection information along with the image. GeoTIFF files embed information about the projection within tags in the .tif or .tiff format file. Images saved using GeoTIFF require only one file with a .tiff or .tif file extension. True GeoTIFF files will import automatically with correct georegistration.

Do not confuse the GeoTIFF format with a different format using .tif files called the "TFW" format. This format uses two files, a .tif file and a=tfw "world" file to provide some georeferencing information. TFW is not the same as GeoTIFF. Adding to the confusion is that some packages will create both a GeoTIFF file as well as a .tfw "world" file. The .tfw file provided in such cases is not part of the GeoTIFF standard.

Manifold looks first for GeoTIFF tags within the image. If these are found the image is imported using GeoTIFF standards. If GeoTIFF tags are not found Manifold checks for a .tfw file and does its best to import using whatever information may be found in that file. If neither GeoTIFF tags nor a .tfw file are found Manifold will import the .tif as an ordinary graphics file without any georegistration.

Limitations of .tfw files

TFW's often will not have all of the information required to georegister an image. For example, some TFW files will only hold offsets and scales (which are correctly read by the importer) but these are not enough to register an image to a location and projection on Earth. For this reason, GeoTIFF is a better standard to use for interchange than TFW.

On occasion it is possible to rescue a TIF/TFW that does not contain complete georegistration information. Many TFW files are saved using Latitude / Longitude values for the pixels in the accompanying .tif image. The actual .tfw file is a plain text file that can be opened with Notepad or some other text editor and inspected to see what it contains. We can sometimes guess by looking at the scales, offsets and dimensions numbers in the TFW that the intended projection system is simple Latitude / Longitude.

If the data is in Latitude / Longitude projection it is fairly easy to import the files and add the necessary projection information by changing the values in the Edit - Assign Projection dialog.

To import a .tif / .tfw in Latitude / Longitude projection:

1. Import the file using the TIF importer.
2. Open the Edit - Assign Projection dialog. The coordinate system will be reported as the default Orthographic projection with offsets and scales from the TFW correctly read into Local Offsets and Local Scales.

3. Write down the values for Local Offsets and Local Scales.

4. Change the projection to Latitude / Longitude.

5. Modify Local Offsets and Local Scales to the values written down in step 3 above.

6. Check to make sure the coordinate system units combo is set to Degree.

7. Press OK.

Note that if one cannot determine the intended projection of the .tfw by inspecting the file there may be some accompanying documentation that describes the intended use of these files.

TIF Files with 16-bit Channels

Image channels are normally 8 bit so that RGB images can be created in the usual way but some TIF files, such as those published by DigitalGlobe that contain satellite images, contain 16 bit image channels.

If a TIF file contains multiple 16-bit channels each channel will be imported into a separate surface component. If the TIF file contains three 16-bit channels the system will create an RGB image component as well. If the TIF file contains four 16-bit channels the system will create an RGBA image component as well.

The composite RGB or RGBA image is synthesized by shrinking each 16-bit channel into an 8-bit range and guessing which channels should be assigned to R, G, B and possibly A. This is bound to result in an unnatural appearance in some cases. If we would like to synthesize an RGB or RGBA image using some other technique or arrangement of channels, we can do this using the full 16-bit surfaces that are also imported.

Although surfaces are normally created with the Display Options shading choice turned on, surfaces created from TIF files will have shading turned off to provide a more "image like" appearance.

EXIF Tags

Exchangeable Image File Format (EXIF) is a standard used mostly by digital cameras for adding information to JPEG images produced by those cameras. EXIF tags can contain a very wide variety of information that is inserted into images by camera programmers. Manifold recognizes both EXIF tags as well as EXIF tag directories written by major camera vendors.

If a GeoTIFF image contains EXIF tags, Manifold will write the information from those tags into the Notes pane properties for that image. To see the EXIF information, open the image: with the focus on the opened image the Notes pane will show the EXIF information for that image.

GeoTIFF images containing orientation EXIF tags will be rotated and flipped as necessary to use the orientation specified.

XMP Tags

Extensible Metadata Platform (XMP) is an XML based metadata format used to embed a seemingly endless variety of information into images. If a GeoTIFF image contains XMP tags Manifold will read the information into the component notes as is done with EXIF tags.

Killing GeoTIFFs with PhotoShop

A GeoTIFF is just a plain, ordinary TIF image with extra tags tossed into the image that specify projection information. That works great if the image editing software you use understands such extra tags. Manifold and certain other image processing packages in the GIS world understand those extra tags. PhotoShop (versions in common usage as of this writing) and virtually all other non-GIS image editing packages don't understand the extra tags. Opening a GeoTIFF with such non-GIS software will almost certainly kill the projection information it contains.

When PhotoShop loads a GeoTIFF it ignores the extra projection system tags and proceeds to display the TIF image just as if it was a plain, ordinary, dumb-format, non-GIS image. When PhotoShop saves that GeoTIFF or
updates it in any way those projection tags have already been lost and the result is that the image is now just a plain, ordinary, dumb-format, non-GIS TIF image. Adios, GeoTIFF!

It is therefore critically important to keep inexpert users away from your GeoTIFF files because they can't seem to resist opening them with some sort of brainless consumer image editing program that will kill the GeoTIFF. Worse yet, it is almost a law of nature that the same sort of inexpert user who does such things will deny to the end of time that they ever touched any files.

The result is that you'll end up with a collection of .tif files that you thought were GeoTIFFs but which you will discover, unpredictably and at random, have been converted into ordinary TIF files and which no longer contain projection information. That's the really annoying part of GeoTIFFs versus TIFs in that since they both end in the same .tif extension there is really nothing to distinguish the two except the presence of invisible projection tags inside the format that only GIS software like Manifold understands.

If you have had the bad luck of inadvertently killing a GeoTIFF by opening it in PhotoShop the bad news is that those projection tags are gone forever. There's nothing about what remains in the .tif that will allow you to get them back. You are now stuck with either getting fresh copies of the GeoTIFFs from the original source, or painfully trying to find out from that original source exactly what the projections were and manually specifying them for each such image using the Edit - Assign Projection dialog.

The solution to the above problems is either to a) keep inexpert users away from your GeoTIFF files, or b) immediately upon acquisition convert the GeoTIFF files to some format, such as ECW, that is less vulnerable to being destroyed by inexpert users. Using ECW is not a bad idea since the files will load a zillion times faster anyway.

World Files and other Formats

In addition to use with TIF files, Manifold supports the use of "world" files to import georegistered images from other formats, specifically from BMP, GIF, JPG, PCX, PNG and TGA files using TFW-like "world" files (with BPW, GFW, GJW, PXW, PGW and TAW extensions respectively).

 Discovery of New TIF Types

TIF and GeoTIFF files can exist in many possible variations, some with very peculiar internal structure known only to their authors. Manifold’s TIF importer handles a very wide range of TIF and GeoTIFF file types, including all commonly occurring types, but it is always possible you may encounter a previously unknown exotic variation. In such cases, the imported image may appear to be blank or all black. If you do encounter a new TIF type and are sure the image does in fact contain data (and is not really blank!), please report the new type to tech@manifold.net so it may be added to the importer.

While all reports of new TIF types are welcome, don't confuse the donation of a new type of TIF with a request for technical support. If you ask any questions or otherwise indicate you would like a response, that's a request for technical support.

If a particular TIF you are working with is not working, the way to get technical support is to use a technical support incident as set forth in the Technical Support topic. If you can’t get a TIF image to appear in a map it is almost certainly a projection error of some kind caused by inaccurate use of the Assign Projection dialog and is almost certainly not a bug in Manifold. Even if it turns out to be a bug in Manifold or the discovery of a new TIF variation (both of which are possible, albeit very rare events), getting an answer back from Technical Support to confirm such a bug or new discovery requires a tech support incident. See the Technical Support topic for details.

Note: A side effect of Manifold’s hunt for new bizarre TIF variations is that users have contributed many examples of TIF files with corrupted headers and other defects. Manifold has therefore evolved the ability to recover from several common cases of corrupted headers and is able in such cases to successfully import the damaged TIF file.

XML Files Created upon Export and Used on Import

Not all formats to which Manifold can export have the ability to correctly save projection information. As a safety measure to ensure that projection information is never lost, Manifold always writes an accompanying .xml file when exporting drawings, images, labels, maps or surface components to a file. The accompanying .xml file contains coordinate system information and some other metadata. The XML file is created for all formats, even those that correctly save coordinate system information.
When Manifold imports drawings, images, labels or surfaces, the system will check for an accompanying .xml file that might have been written by Manifold. If such an .xml file exists, Manifold will read it and use the information it contains to load a correct coordinate system.

See the XML Accessory File Format topic for details on the internal organization of this file.

**Tech Tip**

If you do have a GeoTIFF killed by an inexpert user and that GeoTIFF was created by Manifold, if you still have the auxiliary XML file created by Manifold the situation can be saved. Place the GeoTIFF and its accompanying XML file in the same folder so that when you open the former GeoTIFF the projection information will be restored. Save the GeoTIFF again and the projection information will be re-embedded into what will be once again a GeoTIFF when it is saved. The inclination of inexpert users to kill GeoTIFFs was one of the main reasons Manifold introduced an auxiliary XML file to save projection information, just in case.

**Import Surface**

**Importing Surfaces**

Import surfaces into Manifold using the File - Import - Surface dialog. In addition, surfaces stored within databases as "images" using spatial DBMS technology can be imported by connecting to the database using Database Console, expanding the image to show the Height channel and then importing that Height channel to form an image.

When using the File - Import - Surface dialog the Files of type choice acts as a filter for what is displayed in the browse pane in the Import dialog. Simple file types are imported directly from the Import dialog by choosing the desired type in the Files of type box. More complex imports from data sources such as databases are imported by choosing Data Sources () in the Files of type box and then using the Data Source dialog.

Most surfaces will be imported from file formats, such as USGS DEM, that are dedicated for use as terrain elevation data formats. At times we will also import surfaces from different formats, such as SDTS, which are capable of storing data for surfaces as well as drawings. Some formats, such as .e00, can be used to store drawings, images or surfaces.

Note also that some formats, such as DEM, may occur in different variations that may require use of different formats. For example, 1:250K-scale DEM files use .dem file formats while 1:24K-scale DEM files are really SDTS format files that encode digital elevation data. The former are imported using File - Import - Surface using the DEM importer while the latter are imported using File - Import - Drawing using the SDTS choice.

Some file types, such as ENVI, ERDAS and ERS (ECW) files, can contain either surface or image data, depending on how the data is intended to be interpreted. The ENVI, ERDAS and ERS (ECW) importers occur both in the File - Import - Surface and File - Import - Image menus and have a subtype option to allow specification of image or surface upon import.

Formats used to save surfaces vary widely in their capabilities. Some formats save projection information and some, such as .dxf or many types of .shp files, do not. When a surface format contains projection information, Manifold will automatically use that projection information.

When a surface format does not save projection information, for GIS formats such as ESRI .shp files the surface will be imported into Latitude / Longitude projection as if the units contained were degrees. Most people are sensible enough not to use .shp and similar formats to publish projected maps so this will usually work.

For surfaces imported from non-GIS formats such as AutoCAD .dxf, the surface will be imported using the same default Orthographic projection used for non-geographic images. Such surfaces will always require manual adjustment of parameters in the Edit - Assign Projection dialog.

If a surface format does not save projection information, that information must be added after import via the Edit - Assign Projection dialog.

**Importing Satellite Image Formats**
Many formats used for satellite images are located under File - Import - Surface because such formats are often used to present non-visual data as raster data sets.

For example, HDF, HDF EOS and HDF SeaWiFS formats are found in the File - Import - Surface dialogs. If you don't see a desired format listed in the image importers, check under surfaces.

When importing HDF files, use HDF SeaWiFS for SeaWiFS data in .hdf format and use HDF EOS for Earth Observation Satellite data in .hdf format. While the ordinary HDF converter will also import some data from HDF SeaWiFS files it will ignore multidimensional data chunks (called Scientific Data Sets or SDSs) that are used to contain pixels. Note also there is a CEOS SeaWiFS importer for data in that form.

Importing ESRI exported raster grid files

.e00 is the preferred format for importing grid files from ESRI products. However, ArcInfo can export raster data using "gridsasc" format either as integer or floating point data. There appear to be no standard extensions for such files in universal use within the ESRI community. We suggest saving (or renaming) such files to use either a .grd extension for integer data or an .flt extension for floating point data. Use the File - Import - Surface importer set to GRD files for integer data and FLT files for floating point data.

Typical Surface Formats

Many formats listed as "surfaces" within Manifold could also be considered to be "image" formats. For current information on the following formats, use a good Internet search engine. For example, search "SeaWiFS" for pages describing the SeaWiFS project and data sets.

Interested in support for a new format not listed below? See the Contacting manifold.net topic for information on suggesting a new format.

ADF ESRI .adf format, also sometimes called ArcGrid format. Note that .adf files can contain either raster or vector data. When attempting to import an .adf file as a surface it must contain raster data since surfaces are raster data types. If the .adf file contains vector data it must be imported as a drawing and cannot be imported as a surface.

AVHRR Advanced Very High Resolution Radiometer (AVHRR) data. The AVHRR is a broadband, four or five channel (depending on the model) scanner, sensing in the visible, near-infrared, and thermal infrared bands. Flown on earth observation satellites from TIROS-N to NOAA-15 (orbited 05/13/98). Data available from USGS on CD with resolution of approximately 1.1 km. Get free CDs with intense imagery and EOS sensing that's great for regional observations of vegetation, crop changes.

BIL ESRI .bil files. The ESRI version of what most people call "DEM" files.

CEOS SeaWiFS Sea-viewing Wild Field-of-view Sensor (SeaWiFS). Very cool ocean observation satellite. An eight-band sensor carried on board the SeaStar spacecraft that launched into low Earth orbit on 1 August 1997. 1.1 km resolution. SeaWiFS provides a view of subtle changes in ocean color that signify various types and quantities of marine phytoplankton.

DEM USGS Digital Elevation Module (DEM) files. Note that "real" DEM files are different than DEM data published within SDTS format. To import DEM files published in SDTS format, use the SDTS "drawing" importer.

DEM GLOBE Global Land One-km Base Elevation (GLOBE) project. A 1 km resolution database of terrain elevations for the entire Earth. Perfect for regional terrain elevations such as shaded relief displays for countries or regions. Yet another cool project of the Committee on Earth Observation Satellites (CEOS). Free download via
DEM GTOPO30
GTOPO30 is a global Digital Elevation Model (DEM) with a horizontal grid spacing of 30 arc seconds, approximately 1 kilometer. An older (1996) alternative to GLOBE derived from several raster and vector sources.

DTED
Digital Terrain Elevation Data (DTED) - NIMA military format for terrain elevation. Available in DTED Level 0 resolution for free public download. Higher resolution DTED Level 1 and Level 2 are available to NIMA’s special friends. Manifold reads them all.

EOO
ESRI ArcInfo Export (.e00) format can contain surfaces. If so, Manifold will extract them.

ENVI IMG
Image/surface format from ENVI ("The Environment for Visualizing Images").

ERDAS GIS
ERDAS files in .gis or .lan formats.

ERDAS IMG
ERDAS files in .img format. Works for IMAGINE format .img files.

ERS
ERMapper format used for surfaces. Importing a surface from an ERS file named abc.ERS tries to obtain data from a file named abc.DAT file if the abc file can not be found.

ESRI ASCII GRD
ESRI ASCII format for grids.

ESRI Float Grid FLT
ESRI floating point .flt format (or .hdr format associated with .flt files) for terrain elevation data.

GeoSPOT
A georegistered format for SPOTview data from the SPOT satellite.

GRASS
Surfaces and image in the freeware "GRASS" raster GIS program created by the US Army Corps of Engineers. "70's people find it amusing the Army would call anything it created "grass."

GXF
Geosoft .gxf, an ASCII grid format used for terrain elevation data. Found on the NURE National Uranium Resources USGS CD and in other sources.

HDF

HDF EOS
EOS satellite data in HDF format.

HDF SeaWiFS
SeaWiFS satellite data in HDF format.

IDRISI IMG
IDRISI raster format.

IMDISP
Image Display Program for IDL format.

LAS
Log ASCII Standard (LAS) format used for well logging data. An ASCII format originally developed by the Canadian Well Logging society in the late 1980’s and now used by USGS and others. Choose the .ddr file in the LAS file set.

LULC CTG
Land Use and Land Cover (LULC) data in Composite Theme Grid (CTG) format. Consist of fixed length records with one grid cell for each record. Grid cells are regular point samples of the quad where the center point of each cell is 200 meters from other center points in adjacent cells.

NetCDF
Network Common Data Form data, a format used to describe a very wide variety of data in the scientific
community, using a number of different conventions, or
cantations, as to how that data should be structured
and what it should mean. Manifold's NetCDF importer
makes no effort to interpret data and instead attempts
to import as much of it as possible, in the form of
surfaces or tables. Scalars and 1-dimensional data
modules are imported as tables. 2-dimensional data
modules are imported as surfaces. 3-dimensional data
modules and modules of higher dimensions are
imported as sets of surfaces. The coordinates in the $\text{lat}$
and $\text{lon}$ tables within a NetCDF file are assumed to
refer to the centers of respective pixels for registration
purposes. Components will be registered to the values
in $\text{lat}$ and $\text{lon}$ tables, provided they are uniformly
spaced. Pixels with infinite or invalid floating-point
values will be set invisible on import. Manifold
recognizes and will apply the value of the
$\text{missing_value}$ variable, if present.

**Northwood GRC/GRD**

Northwood / Vertical Mapper grid files containing either
continuous (.grd) or classified (.grc) grids. .grc files
import as palette images since one value is assigned to
all pixels in that class. .grd files import as surfaces since
they provide smoothly-varying gradation.

**NTF**

UK Ordnance Survey NTF files.

**PIX**

Silicon Graphics Alias / Wavefront .pix format

**Raw Text**

Import data from a variety of ASCII files using
configurable user settings.

**Raw Binary**

Import data from a variety of binary files using
configurable user settings.

**SDTS**

SDTS files that contain DEM (Digital Elevation Model)
terrain elevation data. USGS publishes many 1:24K-
scale data sets as SDTS DEM.

**SRTM**

NASA Shuttle Radar Topography Mission data using files
in one or more of the following file formats:

- .hgt - Height. Files import as Int16 surfaces.
- .err - Error data. Files import as Int16 surfaces named
  "...Error".
- .inc - Incidence angle data. Files import as Int16 surfaces
  named "...Incidence".
- .mag - Image. Files import as grayscale images named
  "...Image".
- .pol - Polarization angle. Files import as palette images
  named "...Polarization".

**Surfer GRD**

Surfer 6, 7 or ASCII GRD files, automatically
recognizing the correct Surfer format. Importing a
surface from a Surfer GRD file will derive the local
offset and local scale parameters used for the surface
coordinate system from the file header.

**TAB**

MapInfo .tab files containing surface raster data.

**TIFF**

TIFF rasters containing raster/surface data. Surfaces
imported from .tif files with floating-point values will
automatically set pixels with invalid or infinite floating-
point values invisible.

**XYZ**

Comma-delimited ASCII files containing X, Y and Z
coordinates.

**DB2 Data Sources**

Connect to IBM DB2 equipped with the DB2 Spatial
Extender using IBM’s native spatial connection
technology. A spatial DBMS connection accessed
through the Data Source dialog. This option requires
Enterprise Edition or above.
Oracle Data Sources  Connect to an Oracle data sources using the Oracle Call Interface (OCI), the native Oracle interface for exchanging data. Exchanging images with Oracle data sources via OCI automatically maps images into Oracle GeoRaster form (if GeoRasters are supported by the Oracle product in use). A spatial DBMS connection accessed through the Data Source dialog. This option requires Enterprise Edition or above.

PostgreSQL Data Sources  Connect to PostgreSQL equipped with the PostGIS spatial extender using native PostgreSQL connection technology. Connections made using this data source will have any password required masked so that later usage will not inadvertently expose the password. A spatial DBMS connection accessed through the Data Source dialog. This option requires Enterprise Edition or above.

SQL Server Data Sources  Connect to Microsoft SQL Server 2008 spatial DBMS facilities using native SQL Server 2008 spatial connection technology. A spatial DBMS connection accessed through the Data Source dialog. This option requires Enterprise Edition or above.

Important: Using a vendor's native spatial DBMS connection technology (like those above) provides better performance and allows using features, such as built-in spatial DBMS facilities, not exposed through generic database interfaces such as ODBC, OLE DB and ADO .NET. At the present writing, only Oracle Spatial provides a built-in GeoRaster type. Other spatial DBMS products are supported using Manifold-managed image and surface storage.

Note: because the Database Console provides a more convenient user interface for browsing databases, we will almost always use the Database Console to import or link components from a database instead of using the File - Import or File - Link menu commands.

XML Files Created upon Export and Used on Import

Not all formats to which Manifold can export have the ability to correctly save projection information. As a safety measure to ensure that projection information is never lost, Manifold always writes an accompanying .xml file when exporting drawings, images, labels, maps or surface components to a file. The accompanying .xml file contains coordinate system information and some other metadata. The XML file is created for all formats, even those that correctly save coordinate system information.

When Manifold imports drawings, images, labels or surfaces, the system will check for an accompanying .xml file that might have been written by Manifold. If such an .xml file exists, Manifold will read it and use the information it contains to load a correct coordinate system.

See the XML Accessory File Format topic for details on the internal organization of this file.
Import and Export

Import Surface - ADF
Import ESRI .adf format files. ESRI .adf is sometimes called ArcGrid format.

.adf files can contain raster data, ESRI TIN data or vector (drawing) data. The Manifold .adf importer will automatically configure itself to import either drawings or surfaces as applicable regardless of whether the importer was launched via File - Import - Drawing or via File - Import - Surface. For information on importing .adf files that contain vector data, see the Import Drawing - ADF topic. This topic discusses import of TINs.

Importing a TIN

Importing an .adf file that contains an ESRI TIN pops opens a dialog with the following options:

- **Import points** Import the TIN as a drawing of points, including the height of each point as an attribute in the drawing's table.
- **Import triangles** Import the TIN as a drawing of triangles.
- **Import surface** Create a surface from the TIN of the given Size in pixels (the default).

See Also

Import Drawing - ADF
Import Surface - DEM

DEM files are the most important source of surfaces in the United States. The USGS (www.usgs.gov) publishes vast amounts of data as DEM files for free download from its web site. In addition to the DEM files published by USGS, DEM-style files are used with ESRI's ArcINFO package, with GLOBE worldwide digital elevation data and with GTOPO30 worldwide digital elevation data. All forms of DEM are slightly different and require a different importer.

All DEMS listed below will be imported as georegistered surfaces. Cool!

USGS DEMs

USGS publishes digital elevation data in two resolution levels:

1:250K DEM - 1:250,000-scale Digital Elevation Modules (terrain elevation) suitable for use as surfaces at the county level. These files have a .dem extension. Import them using the DEM Files (*.dem, *.*) setting.

1:24K DEM - 1:24,000-scale Digital Elevation Modules published in SDTS format suitable for use as surfaces at the local level. These correspond to USGS "Quad" maps and are extraordinarily detailed. These files follow SDTS file naming conventions. Use the SDTS drawing importer to import these files (a surface will automatically be created even though a drawing importer was used). Note that unzipping STDS files for a particular USGS Quad will often result in three compressed files. The largest of these files is usually the full accuracy DEM. The others are interpolated to smaller sizes and less detail.

GLOBE

The Global Land One-km Base Elevation (GLOBE) project provides the latest, most modern, public domain terrain elevation database for the entire world. Currently at http://www.ngdc.noaa.gov/seg/topo/globeget.shtml, the data set may be downloaded in 16 "tiles," most of which are very large files (over 100MB each) when unzipped. When unzipped, the GLOBE files have no extensions but will have names like g10g. Import using the DEM GLOBE Files (*.*) setting in File - Import - Surface. GLOBE tiles are so large that most users will crop them to the area of interest using the Crop transform toolbar operator.

GTOPO30

GTOPO30 is a global digital elevation data set with a horizontal grid spacing of 30 arc seconds (approximately 1 kilometer). GTOPO30 was derived from several raster and vector sources of topographic information. GTOPO30 was completed in late 1996. It was developed over a three-year period through a collaborative effort led by staff at USGS’s EROS Data Center. Downloads are currently at http://edcdaac.usgs.gov/gtopo30/gtopo30.html and (after unzipping) may be imported using the DEM GTOPO30 Files (*.dem) setting in File - Import - Surface. Note that although GTOPO30 files use the .dem extension (as well as other file extensions in the GTOPO30 file set) they must be imported using a different importer than the USGS 1:250K-scale DEMs.

GTOPO30 files import into Manifold with invisible pixels at sea level. This makes them handy for use as components in maps showing large areas, since seas may be colored.

ArcINFO DEM

DEM files from ArcINFO are often saved using either a .grd extension or a .bil extension. Import DEM files from ArcINFO using either GRD Files (*.grd) or BIL Files (*.bil, *.hdr) in File - Import - Surface.

Examples
The Alps from GLOBE tile g10, seen as a layer in a map projected into Orthographic projection.

Norwegian fiords from GTOPO30 W020N90 tile seen in a map in Orthographic projection. The surface at left has no background while the one at right has a blue background.
Import Surface - DTED

Using File - Import - Surface with the Files of type filter set to DTED files allows us to import NIMA Digital Terrain Elevation Data (DTED) data. This is a NIMA military format for terrain elevation. DTED data occurs in three resolutions: DTED Level 0 data is published in 30 arc second resolution for free public download. Higher resolution DTED Level 1 and Level 2 are available to NIMA's special friends. Manifold reads them all.

DTED files are georegistered upon import.

Example

Download some DTED data in Military Specification format from the NIMA geospatial engine data server at:

http://geoengine.nima.mil

The data is packaged as a zip file, which will unzip into a dmed file, and dted and text folders. Within the dted folder will be located other folders containing the individual DTED data tiles, usually in files ending in a .dt* extension such as .dt0. Import the .dt* files to create surfaces.

To tile the various surfaces into one big surface, use Copy and Paste to paste them all into one surface.

For example, if in NIMA's server we zoom into the San Francisco Bay area and download the DTED data, after many long minutes of waiting the NIMA server will give us a link to a zip file. We download the zip file, unzip it and find that within the dted folder there are two folders, w122 and w123. Within the w122 folder are four files, n37.avg, n37.dt0, n37.max and n37.min. Within the w123 folder are also four files, also called n37.avg, n37.dt0, n37.max and n37.min. If we import both of the n37.dt0 files (one from the w122 folder and one from the w123 folder), we will end up with two surfaces, one called n37 and the other one called n37 2 (Manifold's default naming scheme appends a "2" to distinguish it from the identically named component already in the project).

The two surfaces show adjacent regions of the San Francisco Bay region. We can copy the n37 2 surface and paste it into the n37 surface to create a single, seamless surface showing the topography of the San Francisco Bay region.
Import and Export

Import Surface - HDF EOS

Although HDF EOS format is found in the importing surfaces menus for Manifold, this format can also be used to store images.

Select All - Check all modules. All data modules will be imported.

Select None - Uncheck all modules. No data modules will be imported. Tech tip: Use this to uncheck all modules before checking those desired.

Select Inverse - Uncheck all previously checked modules and check all previously unchecked modules. This is a fast way to check only a few columns out of many: uncheck those desired and then push Select Inverse.

Modules pane

Check the modules to be imported.

Subtype

Choose between image or surface. Note that one cannot import EOS data that is a surface as an image and vice versa. If one subtype setting does not work, try the other and re-import.

Create Folders

Create folders in the project to host components created from modules.

Tech Tip: Data that arrives in "hdf" files may be any one of the various HDF formats. Try them all if one doesn't work.
Import Surface - Northwood GRC / GRD

The **Import Northwood GRC / GRD File** dialog allows us to import Northwood / Vertical Mapper grid files containing either continuous (.grd) or classified (.grc) grids. *.grc* files import as palette images since one value is assigned to all pixels in that class. *.grd* files import as surfaces since they provide smoothly-varying gradation.

The **Transform Z values with scale and offset parameters** option when checked will transform imported heights using scale and offset parameters stored within the file. By default the option is turned off.
The **Import Raw Text File** dialog allows us to import surfaces from a variety of ASCII files (`.csv`, `.asc`, `.txt`) that provide terrain elevation data with one data point per line and individual fields within the line separated by some delimiting character. The most common form of such files uses a comma character to separate the fields and is thus known as a *comma separated value* file. Such files are often saved with a `.csv` extension but may also be found with other extensions such as `.asc` and `.txt`. This importer is very similar to the XYZ surface importer.

**Type**
Choose the data type represented by the numbers in the file. Complex data types (that is, a pair of floating point values) will be automatically converted to a single floating point value of equivalent precision.

**Channels**
Number of channels.

**Width**
Horizontal (East / West) dimension of resulting surface.

**Height**
Vertical (North / South) dimension of resulting surface.

**Skip**
Number of lines to skip at the beginning of the file before data begins. Used to ignore titles, comment blocks and other prefatory remarks.

**Interleaving**
Interleaving mode for multi-channel surfaces. Not enabled unless the number of channels is greater than 1.

**Delimiter**
Enter characters that should be considered as delimiters. White space characters (space, tab, end-of-line) are always considered delimiters.

**Null value**
Text string used to represent "no data" in the file. For example, `-9999` is often used to indicate no data for a field.
Import Surface - Raw Binary

The Import Raw Binary File dialog provides a generic importer that can import binary surface data from a very wide variety of different binary data files. Such files may appear with a variety of extensions, such as .bin, .dat or other extensions. ADRG files may be imported using this importer.

**Import Raw Binary File**

![Image of the Import Raw Binary File dialog]

- **Type**: Choose the data type represented by the binary data in the file. Complex data types (that is, a pair of floating point values) will be automatically converted to a single floating point value of equivalent precision.

  The type format box next to the **Type** box contains choices for encoding formats for each of the possible types. For example, integer types can have two possible encodings: Little-Endian (Intel) or Big-Endian (Motorola), and floating point types can be one of ten possible encodings: Cray, Data General, Gould, Hewlett Packard, IBM AIX, IBM MVS, IEEE, IEEE Intel, PRIME, Silicon Graphics, Sun, VAX Ultrix, VAX VMS.

- **Channels**: The number of channels in the file.

- **Width**: Horizontal (East / West) dimension of resulting terrain.

- **Height**: Vertical (North / South) dimension of resulting terrain.

- **Skip**: Number of bytes to skip at the beginning of the file before data begins. Used to ignore unnecessary headers and other prefatory items.

  **Interleaving**: Enabled when the number of channels is greater than 1. Determines channel order within the file:

  - **by band**: File contains all samples for channel 1 followed by all for channel 2 and so on.
  - **by line**: File contains all samples for line 1 of channel 1 followed by all samples for line 1 of channel 2 and so on.
  - **by pixel**: File contains samples from all channels for pixel 1 of line 1 followed by samples from all channels for pixel 2 of line 1 and so on.

- **Line Padding**: Number of bytes to skip after each line.

- **Null value**: Text string used to represent "no data" in the file. For example, -9999 is often used to indicate no data for a field.

For an example using the Raw Binary importer, see the Import a Raw Binary File - NLCD topic.
When importing surfaces using Raw Binary, if there is a similarly-named .xml file in the same folder (for example, importing a Montara.bin raw binary surface file plus a Montara.xml accompanying .xml file in the same folder) Manifold will read the .xml file as well. Manifold can then automatically guess the coordinate system, dimensions and type of the imported surface using metadata found in the .xml file to pre-load choices in the Import Raw Binary File dialog.

.xml files accompanying raw binary surface files are expected to be in the same XML format used when exporting surfaces to raw binary. See the Exporting Surfaces topic.
Import Surface - SRTM

SRTM data is derived from NASA’s Shuttle Radar Topography Mission data, collected on the Space Shuttle STS-99 mission. A shuttle-based radar was used to map virtually the entire land area of Earth between latitudes + and - 60 degrees at resolution of one arc second, with a single elevation pixel covering approximately 30 meters. In addition, a 3 arc second (approximately 90-meter resolution) version of the SRTM data has been prepared.

SRTM data is delivered in up to five files for each region covered. The five files deliver height, error, incidence angle, interferometer image and polarization angle. Most commonly distributed SRTM data is simply the height file. In addition, some US government web sites may distribute portions of the SRTM data (such as SRTM data within the US) converted to various other formats such as ESRI .adf or .bil file formats. The Manifold SRTM importer imports all SRTM files. Because Manifold uses standardized SRTM file names to derive coordinate system information, do not rename SRTM files.

The illustration above shows Mount St. Helens in a terrain view from the N46W123.hgt SRTM 30-meter height data file. Z height has been set to 2.5.

The view in the terrain is from the location marked by a reticle in the surface window seen above. Note that the surface contains some scattered portions of invisible pixels from regions where the SRTM data is missing.

SRTM File Types

|hgt| Height data file. Imports as an Int16 surface using the name of the originating file. |
|err| Error data file. Imports as an Int16 surface named "...Error". |
.inc  Incidence angle data file. Imports as an Int16 surface named "...Incidence". Since incidence angles represent natural shading of the surface as seen from the direction opposite to the satellite, Shading for incidence surfaces is turned off by default.

.mag  Image file. Imports as a grayscale image named "...Image".

.pol  Polarization angle file. Imports as a palette image named "...Polarization*. Gray areas indicate HH polarization and green areas indicate VV polarization.

Coordinate system information for SRTM files is taken from the SRTM filenames, which will be in the form N46W123.hgt (the SRTM 1 arcsecond file showing Mount St. Helens). Therefore, do not rename SRTM files before importing them.

The Manifold SRTM importer will automatically switch between 30-meter and 90-meter resolution mode based on the size of the SRTM file being imported.

**Getting Access to SRTM Data**

Access to SRTM elevation data for the US is fast and easy: download it from USGS web sites. However, the US SRTM data is not as useful as SRTM coverages outside the US.

USGS has long provided terrain elevation data for the United States in much greater resolution than the 30-meter maximum resolution of the SRTM data set, but the availability of 30-meter terrain elevation data for almost the entire world is a real breakthrough for public benefit. 30-meter resolution may not as good as data sets that are available for sale to the public by various private entities and state cartographic monopolies like the UK’s Ordnance Survey, but nonetheless it is a significant improvement over previous public domain data sets. The main interest in SRTM data, therefore, is access to those parts of the SRTM data located outside the US.

Unfortunately, access to the SRTM data set is controlled by NASA and NIMA, who have decided to deny the public full access to this important public data set. NASA and NIMA have made public only those parts of the SRTM data set that are within the borders of the US. Worse still, NASA and NIMA have announced an intention of denying the public access to the 30-meter SRTM data if such data is for a region outside the US. They have announced that the public will be allowed access to only the degraded, 90-meter SRTM data.

As of this writing, NASA and NIMA will allow access to international SRTM data only to selected scientists and other persons who agree with NASA’s and NIMA’s data release politics and who are willing to conform to the repressive, and apparently illegal, constraints imposed on this public data by NASA and NIMA. For example, even if a person is "approved" by NASA and NIMA for access to this public data set, that person must agree to give up his or her First Amendment rights to communicate this public data together with their findings to others.

At manifold.net we believe this is yet another example of how NASA and NIMA go about misusing their funding to illegally deny public access to public data. We believe that the anti-public program for international SRTM data is a classic case of how NASA and NIMA are willing to violate the law to prevent the general public from benefiting from public data.

The SRTM data is not classified and is entirely US in origin. There are no copyrights on the data held by any foreign entity or commercial firm. The data was collected in an unclassified program using public funds and does not appear to qualify in any way for the very limited number of provisions that would allow any agency to deny public access to this public data set. SRTM is considerably lower in resolution than terrain elevation data that is freely sold to the public by agencies such as the UK’s Ordnance Survey or published in innumerable paper maps. We believe that by law, NASA and NIMA must provide access to the full 30-meter SRTM data set to all members of the public.

If you agree, please support the effort to pry SRTM data out of NASA. You can do this by filing a Freedom of Information Act request for 30-meter SRTM data for the international region of your choice. If enough users file FOIA requests for this public data set, it makes it tactically more difficult for NASA and NIMA to deny the FOIA requests or to tell lies about why the data has not been made available to the public as required by law.

**Web Sites**
As always, web sites are subject to change at the whim of the webmaster. The following sites were active as of the date of publication of this Help file:


**http://seamless.usgs.gov/** - Entry page for USGS seamless terrain elevation data. Use it to fetch NED (National Elevation Data) files and SRTM files in various formats.

**http://www.jpl.nasa.gov/srtm/cbanddataproducts.html** - Information page on getting access to international data. Use this as a starter page to find contacts at NASA to who you could discuss filing a Freedom of Information Act request to get access to the 1 arc second international data that NASA and NIMA do not want you to have.

**See Also**

Public Access to Public Data - Manifold's call to action to help pry public data out of repressive agencies. Includes a few tips on making FOIA requests.
Import and Export

Import Surface - XYZ

The Import XYZ File dialog allows us to import surfaces from a variety of ASCII files with the extension .xyz or no extension that provide terrain elevation data with one data point per line and individual fields within the line separated by some delimiting character. The most common form of such files uses a comma character to separate the fields and is thus known as a comma separated value file. The importer is very similar to the Raw Text importer.

This importer will scan the file to be imported to automatically detect the correct values for Type, Width, Height, Delimiter and number of header lines (the Skip parameter). The detected values will be pre-loaded into the dialog.

Type Choose the data type represented by the numbers in the file. Complex data types (that is, a pair of floating point values) will be automatically converted to a single floating point value of equivalent precision.

Width Horizontal (East / West) dimension of resulting surface.

Height Vertical (North / South) dimension of resulting surface.

Skip Number of lines to skip at the beginning of the file before data begins. Used to ignore titles, comment blocks and other prefatory remarks that may occur in header lines.

Delimiter Enter characters that should be considered as delimiters. White space characters (space, tab, end-of-line) are always considered delimiters.

Null value Text string used to represent "no data" in the file. For example, -9999 is often used to indicate no data for a field.

Import Table

Importing Tables

Import tables into Manifold using the File - Import - Table dialog. In addition, tables stored within databases using spatial DBMS technology can be imported by connecting to the database using Database Console, clicking on the table to highlight it and then clicking the Import button in the Database Console toolbar.

When using the File - Import - Table dialog the Files of type choice acts as a filter for what is displayed in the browse pane in the Import dialog. Simple file types are imported directly from the Import dialog by choosing the desired type in the Files of type box. More complex imports from data sources such as databases are imported by choosing Data Sources () in the Files of type box and then using the Data Source dialog.

Tables may be imported into a Manifold project or linked into a project. Importing the table brings the data into Manifold so that it may be stored in the Manifold .map file and managed by Manifold System. Linking the table leaves the data stored in an external file. See the Linking Tables topic for information on linking tables into a Manifold project.
Use **File - Import - Table** to import tables from the following data sources:

**ACCDB**  Microsoft Office 2007 and later.

**CSV**  Comma separated values ASCII files, including files using .txt or .csv or other extensions.

**DB**  Borland Paradox.

**DBF**  dBase and FoxPro.

**DSN**  ODBC file data source that describes an ODBC driver to use. Imported using the Data Source dialog.

**HTML**  Tables in web pages.

**MDB**  Access 97 and later versions of Access .mdb files.

**UDL**  Microsoft Universal Data Link file that describes an OLE DB connection. Imported using the Data Source dialog.

**WKx**  Lotus tables.

**XLS**  Excel tables.

**XLSX**  Microsoft Office 2007 and later.

**ADO.NET Data Sources**  A generic ADO.NET connection allowing use of a .NET .DLL assembly, provider class and connection string (parameters). Imported using the Data Source dialog.

**ADO.NET ODBC Data Sources**  Connect using the standard Microsoft .NET ADO.NET provider for ODBC. Imported using the Data Source dialog.

**ADO.NET OLE DB Data Sources**  Connect using the standard Microsoft .NET ADO.NET provider for OLE DB. Imported using the Data Source dialog.

**ADO.NET Oracle Data Sources**  Connect using the standard Microsoft .NET ADO.NET provider for Oracle. Imported using the Data Source dialog.

**ADO.NET SQL Server Data Sources**  Connect using the standard Microsoft .NET ADO.NET provider for SQL Server. Imported using the Data Source dialog.

**ODBC Data Sources**  Connect to a data source using ODBC drivers installed on this system. Imported using the Data Source dialog.

**OLE DB Data Sources**  Connect to an OLE DB data source using an OLE DB provider installed on this system. Imported using the Data Source dialog.

**DB2 Data Sources**  Connect to IBM DB2 equipped with the DB2 Spatial Extender using IBM’s native spatial connection technology. A spatial DBMS connection accessed through the Data Source dialog. This option requires Enterprise Edition or above.

**Oracle Data Sources**  Connect to an Oracle data sources using the Oracle Call Interface (OCI), the native Oracle interface for exchanging data. A spatial DBMS connection accessed through the Data Source dialog. This option requires Enterprise Edition or above.

**PostgreSQL Data Sources**  Connect to PostgreSQL equipped with the PostGIS spatial extender using native PostgreSQL connection technology. Connections made using this data source will have any password required masked so that later usage will not inadvertently expose the password. A
Import and Export

spatial DBMS connection accessed through the Data Source dialog. This option requires Enterprise Edition or above.

**SQL Server Data Sources**

Connect to Microsoft SQL Server 2008 spatial DBMS facilities using native SQL Server 2008 spatial connection technology. A spatial DBMS connection accessed through the Data Source dialog. This option requires Enterprise Edition or above.

**Important:** Using a vendor's native spatial DBMS connection technology (like those above) provides better performance and allows using features, such as built-in spatial DBMS facilities, not exposed through generic database interfaces such as ODBC, OLE DB and ADO .NET. For example, exchanging data with Oracle data sources via OCI automatically maps geometry columns into Oracle SDO_GEOMETRY data and will likewise map geometry columns into the native geometry types used by DB2, PostgreSQL and SQL Server 2008.

**Note:** because the Database Console provides a more convenient user interface for browsing databases, we will almost always use the Database Console to import or link components from a database instead of using the File - Import or File - Link menu commands.

In general, except when connecting to Oracle data sources, ADO.NET is a much faster connection technology than OLE.DB or ODBC.

**Importing Tables from Text Formats**

The CSV importer is used to import tables from most text formats, even those that do not use commas to separate the fields. The CSV importer may be set to use different characters as a delimiter. See the Import Table - CSV topic.

**Windows Locale Settings and Table Import**

Manifold's table import routines will recognize Windows locale settings for .mdb, .xls, some .dbf files (depending on drivers used) and some other formats and will automatically convert text fields to Unicode when the database file locale is incompatible with current regional options. For example, an .mdb database created with Spanish locale settings will have text fields automatically converted to Unicode if it is imported on a Windows machine employing French settings.

**Importing .dbf**

The .dbf or DBF format in MS-DOS and Windows arises from the dBase database management system. Due to the popularity of dBase in the 1980's this format has been adopted by many applications, including those in UNIX and other operating systems. Unfortunately, implementers of programs that read and write ".dbf" have at times "extended" the .dbf standard in ways that are not used by any version of dBase and which cannot be read and written by programs that are strictly compatible with the .dbf format as supported by Microsoft.

Some products will create and read .dbf with unusual names, even those with spaces in them, depending on the operating system or product involved. However, in Windows it is a violation of .dbf spec to use file names with spaces in them for .dbf files.

Manifold includes a special .dbf importer that attempts to auto-adapt to different varieties of .dbf files that may be encountered. This is especially important because ESRI "shapefiles," one of the most popular legacy GIS formats, use .dbf files to store data attributes. Because ESRI shapefiles created in UNIX will frequently violate Microsoft standards for .dbf, the Manifold .dbf driver will accept a wide range of bizarre violations of Microsoft spec that occur in shapefiles created with ESRI products.

For example, .dbf records that begin with an invalid leading character but which are otherwise readable are treated as valid, and Manifold's .dbf driver will read .dbf files with non-standard file and field names, even those with spaces in them.
One problem that does arise is that on occasion MEMO fields written by Microsoft's FoxPro application will not load correctly. A workaround is to use ODBC or OLE DB to connect to such files instead of the built-in .dbf importer. That is, instead of specifying a .dbf file in the Files of type box when importing a .dbf file or linking to it, we would specify an ODBC or OLE DB connection in the Files of type box and then use the ODBC or OLE DB dialogs to connect to that .dbf file.

To connect to a FoxPro .dbf file we must first install an ODBC driver for FoxPro which is part of MDAC 2.5, available at:


To read the file via OLE DB we must install an OLE DB provider for FoxPro, available from Microsoft by searching their downloads page at:

http://www.microsoft.com/downloads

**Importing from Excel**

In general, the most efficient way of saving and working with database information, such as tables, is to use a database, namely Microsoft's Access DBMS or similar product. However, many people use Excel as a "quick and dirty" program in which tables are stored. This is rarely as good a strategy as keeping DBMS information in a real DBMS, but it is usually acceptable so long as we keep in mind the limitations of Excel as compared to a real DBMS.

Tables may be imported from Excel spreadsheets provided that the spreadsheet does, in fact, contain something that can be identified as a table. When importing from Excel, Manifold uses Microsoft's own "Jet" database and file access engine using Microsoft's Jet XLS driver to read Excel .xls file format. If the spreadsheet contains something Microsoft recognizes as a table, Manifold will be able to read that table as well.

A "table" in an Excel spreadsheet is a group of columns with the name of the column at the top and values within each column.

For example, the Excel spreadsheet shown above contains a table. Three columns are labeled at the top with values proceeding down the columns.
In contrast, the spreadsheet above does not contain a “table” as far as Microsoft software or drivers are concerned. There are many ways of intermixing data in what to the human eye is a tabular format that nonetheless as far as Microsoft software is concerned is not a table.

The rule of thumb for success is to keep it simple: save tables in Excel with only one table per .xls file, with each table organized as shown in the first Excel illustration above. Do not add anything else, such as explanatory comments, extra titles or text, subtotal calculations and so on. If you are having difficulty importing an Excel table into Manifold, try opening Access and importing the Excel table into Access. If Access can't find a table in the spreadsheet then Manifold also will not be able to find a table in that spreadsheet.

One limitation of Excel when used instead of a DBMS is that Excel does not maintain strong data types as would a DBMS. When the Microsoft Jet XLS driver reads in a table from Excel, it must infer the desired data types by examining the first few values it encounters.

For example, if an Excel column contains both text and numeric entries but the first dozen entries in the column are all numeric, the Jet XLS driver may decide that the column should be treated as a numeric column and might therefore import all values as if they were representations, say, of Floating Point Double numbers instead of a mixture of text and numbers.

The classic situation in which trouble occurs is the case of Excel tables with columns that contain ZIP codes. ZIP codes are postal codes in the United States that, despite appearing as numbers, must be treated as text since leading zeros are significant. For example, there are ZIP codes in the US such as "02138" (in Cambridge, Massachusetts) in which the leading zero is significant. Much software that utilizes ZIP codes will reject the numeric equivalent of "2138" as a malformed ZIP code because it does not contain five characters.

If we store ZIP codes in a column in an Excel table, upon import into Microsoft Access or any other program (such as Manifold) that uses Microsoft's Jet XLS driver to read the .xls file the column containing ZIP codes will be imported as a numeric column and thus leading zeros will be truncated.

This situation may be remedied by either storing such DBMS information in a true DBMS, like Access, or if we must use Excel tables by manipulating it after import into Manifold. For example, such a ZIP code column that was imported as a numeric column could be changed into a text column, all values with only four digits identified and then a leading "0" added to them under the assumption that the reason they are four digits is because a leading "0" was truncated.

Another method is to add ten or so “dummy” values at the beginning of the table where each of the ZIP code values is alphabetic text, thus forcing the Jet XLS driver to recognize that column as text and so preserve any leading zero characters. After the table is imported into Manifold, we can delete the dummy values at the beginning of the table.

**Regional Settings**

When importing data from text file formats like .csv or from .dbf or shapefiles the Regional Options in the system have to match settings within the file that is being imported. This is because such simple formats are too stupid to
understand that in different countries people use different symbols to denote decimal points (that is, either a dot character or a comma). To change Regional Options, do the following:

- Go to the Control Panel and open the Regional Options applet.
- Go to the Numbers page,
- Ensure that the Decimal Symbol is set to ‘.’ (dot),
- Ensure that the List Separator Symbol is set to ‘,’ (comma),
- Press Apply to apply changes if there were any,
- Import the .csv file or shapefile desired.
- After import, restore the original settings of Decimal Symbol and List Separator Symbol if you like.

Another option is to invoke the Data Sources applet (hidden within Administrative Tools folder of the Control Panel if you’re running Win2K) and create a file DSN for the .csv file you want to import. After creating the DSN configure it to use the comma character as a data separator.

Important Note when Using 64-bit Manifold Editions

Due to a lack of required Microsoft facilities in 64-bit Windows systems, Manifold in 64-bit mode cannot export, import, export or link to DB, HTML, MDB, XLS or WKx format files. This includes no access to the MDB parts of Manifold MFD and MapInfo TAB imports. The workaround for importing or exporting such files is to launch Manifold in 32-bit mode by using the Manifold System (32-bit) shortcut, perform the export from or import into a .map project file and then re-launch Manifold in 64-bit mode using the Manifold System (64-bit) shortcut. Linking is more complex: the data must be in some format usable in 64-bit mode within 64-bit Windows systems, such as a SQL Server database. Alternatively, the data can be kept within a Manifold .map project file and linked using the Manifold ODBC driver.

See Also

Importing and Linking Tables
Import Table - CSV
Import Table - CSV

CSV refers to a family of ASCII (plain text) formats that save tabular data in the form of one line per record with the values for each field in that record separated by a delimiting character. The comma is most frequently used as a delimiter, hence the name Comma Separated Value (CSV) format. Files that use characters other than commas, such as tab characters or some other character, are not "CSV" files as a matter of strict nomenclature and often use a different three letter extension such as .txt or .tab. Manifold's CSV importer can handle most such cases.

There is considerable variation in how different programs work with "CSV" format. Some programs like to see a comma at the end of each line and others do not. Most programs expect to see only string field values enclosed in quotes, but some like to see all values enclosed in quotation marks whether they are strings or not. Manifold's CSV exporter provides options to deal with both situations.

To import a table from .csv format:

1. Make sure the file to be imported is located on read/write media such as your hard disk, and that you have write permissions in the folder in which the file is located.
2. Choose File - Import - Table from the main menu.
3. In the Import dialog choose CSV Files in the Files of type box and navigate to the file desired. Double-click on the file or click on it and press Open.
4. In the Import CSV File dialog, choose the List Delimiter character and other options. Check the First line contains field names box if the first line in the file contains field names.
5. Press OK.

Controls

- **List delimiter** Enter that character that is used as the delimiter between fields, or choose from a pull down list of frequently used characters. Manifold will suggest a default character, such as a comma, based on an examination of the first few lines of the file.

- **Decimal delimiter** Enter that character that is used as the decimal point delimiter, or choose from a pull down list of frequently used characters. Manifold will suggest a default character, such as a period (in the US) or a comma (in Europe), based on an examination of the first few lines of the file.

- **Text qualifier** Enter that character that is used to denote text fields, or choose from a pull down list of frequently used characters. Manifold will suggest a default character, such as a quotation mark, based on an examination of the first few lines of the file.

- **First line contains field names** Some CSV files contain the names of the fields in the first line. Check this box if that is the case for the file being imported.

- **Force ANSI character set** If unchecked (the default) text data is interpreted using the locale of the current user. If checked, text data is interpreted as using the ANSI code page.

- **Import all fields as text** Import all fields, including numeric fields, as text fields.

- **Scan first … lines to determine field type** Manifold can scan the given number of lines to guess what should be used as a field type for most fields. If the first few lines are likely to be blank or to contain non-representative data, use more lines. For example, suppose a given field might contain values like "123", "333" or "Suite B". If in the first few lines only obvious numeric types are encountered Manifold will guess the field should be a numeric field. If a larger number of lines are scanned Manifold might encounter a "Suite B" type value and thus realize that the field should be a text field. This helps resolve confusion in cases where quotes are
not used within the CSV file to distinguish text fields.

(preview pane) Previews the effect of the specified import options. If what is seen in this pane doesn't make sense, adjust the import options such as the List delimiter or other options.

CSV and similar text formats lose fine control over how numeric fields are represented. Values end up being either text fields or numbers. A better choice for data interchange in the Microsoft world is to use Microsoft's .mdb format whenever possible.

If desired, we can manually specify the types of fields after the table is imported. For example, we could import all fields as text and then manually specify different fields to be integers, floating point numbers and so on as desired.

See Also

Export Table - CSV

Link Table

Linking Tables

Link tables into Manifold using the File - Link - Table dialog. In addition, tables stored within databases using spatial DBMS technology can be linked by connecting to the database using Database Console, clicking on the table to highlight it and then clicking the Link button in the Database Console toolbar.

When using the File - Link - Table dialog the Files of type choice acts as a filter for what is displayed in the browse pane in the Link dialog. Simple file types are linked directly from the Link dialog by choosing the desired type in the Files of type box. More complex links from data sources such as databases are linked by choosing Data Sources () in the Files of type box and then using the Data Source dialog.

Tables may be imported into a Manifold project or linked into a project. Importing the table brings the data into Manifold so that it may be stored in the Manifold .map file and managed by Manifold System. Linking the table leaves the data stored in an external file. See the Importing Tables topic for information on importing tables into a Manifold project.

Linking a table into a project leaves the table's data stored in an external database file that is managed by a database provider other than Manifold. Linked tables may be created from external data sources using the following formats and data sources:

- **ACCDB** Microsoft Office 2007 and later. Can be linked read/write.
- **CSV** Comma separated values ASCII files, including files using .txt or .csv or other extensions.
- **DB** Borland Paradox.
- **DBF** dBase and FoxPro.
- **DSN** ODBC file data source that describes an ODBC driver to use. Linked using the Data Source dialog.
- **HTML** Tables in web pages.
- **MDB** Access 97 and later versions of Access .mdb files.
- **UDL** Microsoft Universal Data Link file that describes an OLE DB connection. Linked using the Data Source dialog.
- **WKx** Lotus tables.
- **XLS** Excel tables.
XLSX  Microsoft Office 2007 and later. Can be linked read-only.

**ADO.NET Data Sources**
A generic ADO.NET connection allowing use of a .NET .DLL assembly, provider class and connection string (parameters). Linked using the Data Source dialog.

**ADO.NET ODBC Data Sources**
Connect using the standard Microsoft .NET ADO.NET provider for ODBC. Linked using the Data Source dialog.

**ADO.NET OLE DB Data Sources**
Connect using the standard Microsoft .NET ADO.NET provider for OLE DB. Linked using the Data Source dialog.

**ADO.NET Oracle Data Sources**
Connect using the standard Microsoft .NET ADO.NET provider for Oracle. Linked using the Data Source dialog.

**ADO.NET SQL Server Data Sources**
Connect using the standard Microsoft .NET ADO.NET provider for SQL Server. Linked using the Data Source dialog.

**ODBC Data Sources**
Connect to a data source using ODBC drivers installed on this system. Linked using the Data Source dialog.

**OLE DB Data Sources**
Connect to an OLE DB data source using an OLE DB provider installed on this system. Linked using the Data Source dialog.

**DB2 Data Sources**
Connect to IBM DB2 equipped with the DB2 Spatial Extender using IBM's native spatial connection technology. A spatial DBMS connection accessed through the Data Source dialog. This option requires Enterprise Edition or above.

**Oracle Data Sources**
Connect to an Oracle data sources using the Oracle Call Interface (OCI), the native Oracle interface for exchanging data. A spatial DBMS connection accessed through the Data Source dialog. This option requires Enterprise Edition or above.

**PostgreSQL Data Sources**
Connect to PostgreSQL equipped with the PostGIS spatial extender using native PostgreSQL connection technology. Connections made using this data source will have any password required masked so that later usage will not inadvertently expose the password. A spatial DBMS connection accessed through the Data Source dialog. This option requires Enterprise Edition or above.

**SQL Server Data Sources**
Connect to Microsoft SQL Server 2008 spatial DBMS facilities using native SQL Server 2008 spatial connection technology. A spatial DBMS connection accessed through the Data Source dialog. This option requires Enterprise Edition or above.

**Important:** Using a vendor's native spatial DBMS connection technology (like those above) provides better performance and allows using features, such as built-in spatial DBMS facilities, not exposed through generic database interfaces such as ODBC, OLE DB and ADO .NET. For example, exchanging data with Oracle data sources via OCI automatically maps geometry columns into Oracle SDO_GEOMETRY data and will likewise map geometry columns into the native geometry types used by DB2, PostgreSQL and SQL Server 2008.

**Note:** because the Database Console provides a more convenient user interface for browsing databases, we will almost always use the Database Console to import or link components from a database instead of using the File - Import or File - Link menu commands.

In general, **ADO.NET** is a much faster connection technology than **OLE.DB** or **ODBC**.
Linking to a Table within the Project Not Allowed

Although Manifold can link a drawing, image or surface to a table within the same project, linking a table to a different table in the same project is not allowed. This prevents circular references (tables that link to each other in a circular loop) from happening.

Important Note when Using 64-bit Manifold Editions

Due to a lack of required Microsoft facilities in 64-bit Windows systems, Manifold in 64-bit mode cannot export, import, export or link to DB, HTML, MDB, XLS or WKX format files. This includes no access to the MDB parts of Manifold MFD and MapInfo TAB imports. The workaround for importing or exporting such files is to launch Manifold in 32-bit mode by using the Manifold System (32-bit) shortcut, perform the export from or import into a .map project file and then re-launch Manifold in 64-bit mode using the Manifold System (64-bit) shortcut.

Linking is more complex: the data must be in some format usable in 64-bit mode within 64-bit Windows systems, such as a SQL Server database. Alternatively, the data can be kept within a Manifold .map project file and linked using the Manifold ODBC driver.

See Also

Importing and Linking Tables for more detailed discussion.

Link Drawing

Link Drawings

Link drawings into Manifold using the File - Link - Drawing dialog. In addition, drawings stored within databases using spatial DBMS technology can be linked by connecting to the database using Database Console, clicking on the drawing to highlight it and then clicking the Link button in the Database Console toolbar.

The File - Link - Drawing dialog always launches the Data Source dialog to enable choice of a data source from which the drawing will be linked. The Data Source dialog is used even in the case of linking drawings from databases stored in simple file types, such as .mdb. See the Data Source dialog topic for an example of creating a data source for a simple file type.

Manifold can create a linked drawing from data stored in tables or queries, using either geometry information stored in the table or latitude / longitude data in columns (geocoded tables). Manifold can also link drawings from spatial DBMS storage.

Linked drawings using geometry in tables are fully editable, are read/write and can include areas, lines or points. When using simple geocoded tables, linked drawings are read-only and by default consist of points, one point for each record, but may optionally be configured to create lines as well.

Linked drawings can be created from tables using any of the data access methods provided by Manifold for use in the Data Source dialog, such as connection to an ADO.NET, ODBC or OLE DB data source. When connecting to vendor-provided spatial DBMS, such as DB2 with IBM Spatial Extender, Oracle, PostgreSQL or SQL Server 2008, Manifold can connect using the vendor's native spatial DBMS connection technology for enhanced performance. Enterprise Edition is required to connect to a native spatial DBMS using the vendor's native connection technology, for example, to connect to Oracle using Oracle's OCI.

Linking to a Table within the Project

Manifold can also link a drawing to a table within the same project. Choose File - Link - Drawing and then use This Project in the Data Source dialog. See the example in the Linked Drawings from Geocoded Tables topic.

Important Note when Using 64-bit Manifold Editions

Due to a lack of required Microsoft facilities in 64-bit Windows systems, Manifold in 64-bit mode cannot export, import, export or link to DB, HTML, MDB, XLS or WKX format files. This includes no access to the MDB parts of Manifold MFD and MapInfo TAB imports. The workaround for importing or exporting such files is to launch Manifold in 32-bit mode by using the Manifold System (32-bit) shortcut, perform the export from or import into a
.map project file and then re-launch Manifold in 64-bit mode using the Manifold System (64-bit) shortcut.

Linking is more complex: the data must be in some format usable in 64-bit mode within 64-bit Windows systems, such as a SQL Server database. Alternatively, the data can be kept within a Manifold .map project file and linked using the Manifold ODBC driver.

See Also

Importing and Linking Tables
Linked Drawings
Linked Drawings from Geocoded Tables
Create a Linked Drawing from a Geocoded Table
Geometry in Tables
Spatial DBMS

Link Image

Linking Images

Link images into Manifold using the File - Link - Image dialog. In addition, images stored within databases using spatial DBMS technology can be linked by connecting to the database using Database Console, clicking on the image to highlight it and then clicking the Link button in the Database Console toolbar.

When using the File - Link - Image dialog the Files of type choice acts as a filter for what is displayed in the browse pane in the Link dialog. Simple file types and dedicated server options are linked directly from the Link dialog by choosing the desired type in the Files of type box. More complex links from data sources such as databases are linked by choosing Data Sources () in the Files of type box and then using the Data Source dialog.

Images may be imported into a Manifold project or linked into a project. Importing the image brings the data into Manifold so that it may be stored in the Manifold .map file and managed by Manifold System. Linking the image leaves the data stored in an external file or acquired dynamically from an external provider. See the Linked Images topic for important information on using linked images.

Manifold can link images from the following sources:

- **CSV**  Comma separated values ASCII files, including files using .txt or .csv or other extensions. Linked using the Data Source dialog.
- **DB**   Borland Paradox. Linked using the Data Source dialog.
- **DBF**  dBase and FoxPro. Linked using the Data Source dialog.
- **DSN**  ODBC file data source that describes an ODBC driver to use. Linked using the Data Source dialog.
- **ECW / JPEG2000**  Compressed image in either ECW or JPEG2000 compressed format. Any size ECW or JPEG2000 image may be imported.
- **HTML**  Tables in web pages. Linked using the Data Source dialog.
- **MDB**  Access 97 and later versions of Access .mdb files. Linked using the Data Source dialog.
- **UDL**  Microsoft Universal Data Link file that describes an OLE DB connection. Linked using the Data Source dialog.
- **WKx**  Lotus tables. Linked using the Data Source dialog.
- **XLS**  Excel tables. Linked using the Data Source dialog.
- **Image Library Files**  An image library is a folder that contains one or more individual image files, called image tiles, that together
from the desired image.

**ADO.NET Data Sources**
A generic ADO.NET connection allowing use of a .NET .DLL assembly, provider class and connection string (parameters). Linked using the Data Source dialog.

**ADO.NET ODBC Data Sources**
Connect using the standard Microsoft .NET ADO.NET provider for ODBC. Linked using the Data Source dialog.

**ADO.NET OLE DB Data Sources**
Connect using the standard Microsoft .NET ADO.NET provider for OLE DB. Linked using the Data Source dialog.

**ADO.NET Oracle Data Sources**
Connect using the standard Microsoft .NET ADO.NET provider for Oracle. Linked using the Data Source dialog.

**ADO.NET SQL Server Data Sources**
Connect using the standard Microsoft .NET ADO.NET provider for SQL Server. Linked using the Data Source dialog.

**ECWP Image Servers**
Streaming image server providing compressed images in ECW format.

**Manifold Image Servers**
Link images from image servers connected by ISI drivers to Manifold, such as Manifold Image Servers or third party servers such as Google, Yahoo! or Microsoft Virtual Earth. Use of third party servers requires installation of open source ISI drivers for Google servers. See the Linked Images from Manifold Image Servers topic for details.

**ODBC Data Sources**
Connect to a data source using ODBC drivers installed on this system. Linked using the Data Source dialog.

**OLE DB Data Sources**
Connect to an OLE DB data source using an OLE DB provider installed on this system. Linked using the Data Source dialog.

**OGC WMS Servers**
Streaming image server using well-known OGC WMS protocols.

**TerraServer Server**
Link an image from Microsoft's TerraServer image server.

**DB2 Data Sources**
Connect to IBM DB2 equipped with the DB2 Spatial Extender using IBM's native spatial connection technology. A spatial DBMS connection accessed through the Data Source dialog. This option requires Enterprise Edition or above.

**Oracle Data Sources**
Connect to an Oracle data sources using the Oracle Call Interface (OCI), the native Oracle interface for exchanging data. Exchanging images with Oracle data sources via OCI automatically maps images into Oracle GeoRaster form (if GeoRasters are supported by the Oracle product in use). A spatial DBMS connection accessed through the Data Source dialog. This option requires Enterprise Edition or above.

**PostgreSQL Data Sources**
Connect to PostgreSQL equipped with the PostGIS spatial extender using native PostgreSQL connection technology. Connections made using this data source will have any password required masked so that later usage will not inadvertently expose the password. A spatial DBMS connection accessed through the Data Source dialog. This option requires Enterprise Edition or above.

**SQL Server Data Sources**
Connect to Microsoft SQL Server 2008 spatial DBMS facilities using native SQL Server 2008 spatial connection technology. A spatial DBMS connection accessed through the Data Source dialog. This option requires Enterprise Edition or above.

**Important:** Using a vendor's native spatial DBMS connection
technology (like those above) provides better performance and allows using features, such as built-in spatial DBMS facilities, not exposed through generic database interfaces such as ODBC, OLE DB and ADO.NET. At the present writing, only Oracle Spatial provides a built-in GeoRaster type. Other spatial DBMS products are supported using Manifold-managed image and surface storage.

**Note:** because the Database Console provides a more convenient user interface for browsing databases, we will almost always use the Database Console to import or link components from a database instead of using the File - Import or File - Link menu commands.

In general, **ADO.NET** is a much faster connection technology than **OLE.DB** or **ODBC**.

**Imported or Linked Images from Tables or Queries**

Manifold can import images or link images from information stored in tables or provided by queries. The tables and queries used to create linked images may be inside the Manifold project or they may be in external databases.

Linked images may also be created in a two step process, where the virtual table for an existing image is used in a query to manipulate that image or to fetch part of the image based upon desired criteria. A linked image can then be created from that query. This technique is often used within Manifold projects that will be used in an IMS website.

Linked images may also be created from tables or queries stored in external databases. For example, we could use a query to select all columns for all pixels in an image's virtual table and then export that query as a table into an .mdb file or to some other database storage.

When an image is linked from a table or query, each record in the table or query produces a pixel. The coordinates of the pixel are taken from the X and Y columns (cast to integer) specified in the Link dialog. The color of the pixel is taken from either the Color column or the channel columns.

If you have Manifold System Enterprise Edition or greater you can also import images or link images from Oracle data sources that use Oracle's Spatial or Locator technologies. Importing images from Oracle data sources is virtually identical to linking them from Oracle data sources. See the **Linked Images from Oracle Servers** topic.

**Linking to a Table within the Project**

Manifold can also link an image to a table within the same project. In the Link dialog choose Data Sources () and then use This Project in the Data Source box.

**Converting a Linked Image to a Local Image**

An image linked from a remote server can be converted to a native (local) image within the project using the Image - Convert To or Image - Unlink commands. Images can also be fetched with different levels available using the Image - Download command.

**Important Note when Using 64-bit Manifold Editions**

Due to a lack of required Microsoft facilities in 64-bit Windows systems, Manifold in 64-bit mode cannot export, import, export or link to **DB**, **HTML**, **MDB**, **XLS** or **WKx** format files. This includes no access to the **MDB** parts of Manifold **MFD** and MapInfo **TAB** imports. The workaround for importing or exporting such files is to launch Manifold in 32-bit mode by using the Manifold System (32-bit) shortcut, perform the export from or import into a .map project file and then re-launch Manifold in 64-bit mode using the Manifold System (64-bit) shortcut. Linking is more complex: the data must be in some format usable in 64-bit mode within 64-bit Windows systems, such as a SQL Server database. Alternatively, the data can be kept within a Manifold .map project file and linked using the Manifold ODBC driver.

**See Also**
Compressed Images
Linked Images
Linked Images from Google Servers
Linked Images from OGC WMS Servers
Linked Images from Oracle Servers
Linked Images from TerraServer
Virtual Tables for Images and Surfaces
Raster Extensions
Queries and Images or Surfaces

Link Surface

Linking Surfaces

Link surfaces into Manifold using the File - Link - Surface dialog. The File - Link - Surface dialog always launches the Data Source dialog to enable choice of a data source from which the surface will be linked. The Data Source dialog is used even in the case of linking surfaces from databases stored in simple file types, such as .mdb. See the Data Source dialog topic for an example of creating a data source for a simple file type.

Manifold can create a linked surface from data stored in an external table or query, the records of which contain position and height values for the surface's pixels. If the data in the table or query changes, such as when records are added, deleted, or their height values are changed, the corresponding pixels in the surface will automatically be added, deleted or changed in height. Linked surfaces are read-only because their content is controlled entirely by the external table or query. To make a change in a linked surface, change the data in the external table or returned by the query.

Linked surfaces are therefore perfect for a wide class of Manifold IMS applications where surfaces must be dynamically updated based on changing information stored in database management systems. For example, a pollution emissions tracking application might show the current shape and density contouring of a surface representing a pollutant cloud based upon data that is stored in a SQL Server database. To take another example, a layer in an IMS application may be created that shows that part of a surface that is above a particular height, so that users viewing a map can see what parts of the map lie above or below a particular height.

Linked surfaces may be created from tables or queries using the following formats and data sources, connecting through the Data Source dialog:

- CSV Comma separated values ASCII files, including files using .txt or .csv or other extensions.
- DB Borland Paradox.
- DBF dBase and FoxPro.
- DSN ODBC file data source that describes an ODBC driver to use.
- HTML Tables in web pages.
- MDB Access 97 and later versions of Access .mdb files.
- UDL Microsoft Universal Data Link file that describes an OLE DB connection.
- WKx Lotus tables.
- XLS Excel tables.
- ADO.NET Data Sources A generic ADO.NET connection allowing use of a .NET .DLL assembly, provider class and connection string (parameters).
- ADO.NET ODBC Data Connect using the standard Microsoft .NET ADO.NET
Import and Export

Sources provider for ODBC.

**ADO.NET OLE DB Data Sources**
Connect using the standard Microsoft .NET ADO.NET provider for OLE DB.

**ADO.NET Oracle Data Sources**
Connect using the standard Microsoft .NET ADO.NET provider for Oracle.

**ADO.NET SQL Server Data Sources**
Connect using the standard Microsoft .NET ADO.NET provider for SQL Server.

**ODBC Data Sources**
Connect to a data source using ODBC drivers installed on this system.

**OLE DB Data Sources**
Connect to an OLE DB data source using an OLE DB provider installed on this system.

**Oracle Data Sources**
Connect to an Oracle data sources using the Oracle Call Interface (OCI), the native Oracle interface for exchanging data. Manifold can **import** surfaces from Oracle when surfaces are stored as GeoRasters in Oracle, but can **link** surfaces only from a table or query in Oracle using a combination of coordinate and height columns.

**Note:** because the Database Console provides a more convenient user interface for browsing Oracle databases, we will almost always use the Database Console to import or link components from an Oracle database instead of using the File - Import or File - Link menu commands. This option requires Enterprise Edition or above. See the Oracle Spatial Facilities topic for additional information.

**This Project**
Link a surface to a table within this project.

In general, **ADO.NET** is a much faster connection technology than **OLE.DB** or **ODBC**.

**Linking to a Table within the Project**

Manifold can also link a surface to a table within the same project. In the Link dialog choose **Data Sources ()** and then use **This Project** in the **Data Source** box.

**Important Note when Using 64-bit Manifold Editions**

Due to a lack of required Microsoft facilities in 64-bit Windows systems, Manifold in 64-bit mode cannot export, import, export or link to **DB**, **HTML**, **MDB**, **XLS** or **WKx** format files. This includes no access to the **MDB** parts of Manifold **MFD** and MapInfo **TAB** imports. The workaround for importing or exporting such files is to launch Manifold in 32-bit mode by using the **Manifold System (32-bit)** shortcut, perform the export from or import into a **.map** project file and then re-launch Manifold in 64-bit mode using the **Manifold System (64-bit)** shortcut. Linking is more complex: the data must be in some format usable in 64-bit mode within 64-bit Windows systems, such as a SQL Server database. Alternatively, the data can be kept within a Manifold **.map** project file and linked using the Manifold ODBC driver.

**See Also**

**Linked Surfaces**
**Virtual Tables for Images and Surfaces**
**Raster Extensions**
**Queries and Images or Surfaces**
Export Drawing

Exporting Drawings

Export drawings from Manifold by right-clicking on the drawing in the project pane and choosing export, or by opening the drawing and then using the **File - Export - Drawing** dialog. In addition, drawings can be exported into databases using spatial DBMS technology can be exported by using the Data Source dialog. See the Export Drawing - Oracle topic for a typical example of exporting to a spatial DBMS as can also be done for DB 2 with Spatial Extender, PostgreSQL/PostGIS, SQL Server 2008, SQL Server 2005 with Manifold Spatial Extender, and just about any DBMS using Manifold's ability to create and use generic spatial indices.

When using the **File - Export - Drawing** dialog the **Files of type** choice acts as a filter for what is displayed in the browse pane in the Export dialog. Simple file types are exported directly from the Export dialog by choosing the desired type in the Files of type box. More complex exports into data sources such as databases are exported by choosing Data Sources in the Files of type box and then using the Data Source dialog.

Manifold System exports drawings directly into a selected set of the most common GIS interchange formats. Enterprise Edition exports to all Professional Edition formats as well as adding .e00 and export to spatial DBMS formats such as Oracle spatial database formats.

**Professional Edition**

- **DXF** AutoCAD 14 and greater .dxf format.
- **MIF** MapInfo .mid / .mif format.
- **KML, KMZ** XML-based KML or KMZ format used by Google Earth and others.
- **MFD/MDB** Manifold Release 4.50 .mfd / .mdb format.
- **SDTS** US Federal SDTS (Spatial Data Transfer Standard) format.
- **SHP** ESRI ArcView .shp ("shapefile") format.

**Enterprise Edition**

- **E00** ESRI ArcInfo .e00 format.

**DB2 Data Sources** Connect to IBM DB2 equipped with the DB2 Spatial Extender using IBM's native spatial connection technology. A spatial DBMS connection accessed through the Data Source dialog. This option requires Enterprise Edition or above.

**Oracle Data Sources** Connect to an Oracle data sources using the Oracle Call Interface (OCI), the native Oracle interface for exchanging data. A spatial DBMS connection accessed through the Data Source dialog. This option requires Enterprise Edition or above.

**PostgreSQL Data Sources** Connect to PostgreSQL equipped with the PostGIS spatial extender using native PostgreSQL connection technology. Connections made using this data source will have any password required masked so that later usage will not inadvertently expose the password. A spatial DBMS connection accessed through the Data Source dialog. This option requires Enterprise Edition or above.

**SQL Server Data Sources** Connect to Microsoft SQL Server 2008 spatial DBMS facilities using native SQL Server 2008 spatial connection technology. A spatial DBMS connection accessed through the Data Source dialog. This option requires Enterprise Edition or above.

**Important:** Using a vendor's native spatial DBMS connection technology (like those above) provides better performance and allows using features, such as built-in spatial DBMS facilities, not exposed through generic database interfaces such as ODBC, OLE DB and ADO.
.NET. For example, exchanging data with Oracle data sources via OCI automatically maps geometry columns into Oracle SDO_GEOMETRY data and will likewise map geometry columns into the native geometry types used by DB2, PostgreSQL and SQL Server 2008.

See the links above for topics on each exporter.

To export a drawing:

1. Open the drawing.
2. Choose File - Export - Drawing.
3. In the Export dialog choose the desired format in the Save as type box.
4. Browse to the desired folder and specify a name (without extension) in the File name box.
5. Press OK.

Limitations of Target Formats

Drawing formats in common use in GIS and CAD do not usually have the ability to host the full set of information content available within Manifold drawings. Most exports therefore will simplify data in some way. Field names may need to be truncated to eight characters, or projection information may need to be abandoned. Data attributes may not be able to store the full range of data types allowed within Manifold. Formats will not be preserved except with colors in .mif export.

If a drawing is to be exported to a legacy GIS format, such as .shp format, it should be maintained with the limitations of the target format in mind. A least common denominator set of guidelines:

- Use simple field names, no more than eight characters.
- Avoid spaces or other special characters in drawing or field names.
- Use only simple data types: integer, floating point, or text.
- Assume any opacity settings will be lost.
- Assume only default formats will apply.

The above limitations are profound, of course, and negate some of the benefits of using Manifold.

XML Files Created upon Export and Used on Import

Not all formats to which Manifold can export have the ability to correctly save projection information. As a safety measure to ensure that projection information is never lost, Manifold always writes an accompanying .xml file when exporting drawings, images, labels, maps or surface components to a file. The accompanying .xml file contains coordinate system information and some other metadata. The XML file is created for all formats, even those that correctly save coordinate system information.

When Manifold imports drawings, images, labels or surfaces, the system will check for an accompanying .xml file that might have been written by Manifold. If such an .xml file exists, Manifold will read it and use the information it contains to load a correct coordinate system.

See the XML Accessory File Format topic for details on the internal organization of this file.

Important Note when Using 64-bit Manifold Editions

Due to a lack of required Microsoft facilities in 64-bit Windows systems, Manifold in 64-bit mode cannot export, import, export or link to DB, HTML, MDB, XLS or WKx format files. This includes no access to the MDB parts of Manifold MFD and MapInfo TAB imports. The workaround for importing or exporting such files is to launch Manifold in 32-bit mode by using the Manifold System (32-bit) shortcut, perform the export from or import into a .map project file and then re-launch Manifold in 64-bit mode using the Manifold System (64-bit) shortcut. Linking is more complex: the data must be in some format usable in 64-bit mode within 64-bit Windows systems, such as a SQL Server database. Alternatively, the data can be kept within a Manifold .map project file and linked using the Manifold ODBC driver.
See Also

Spatial DBMS
Oracle Spatial Facilities
Example: Storing a Drawing in Oracle
Manifold Spatial DBMS Facilities
Example: Storing a Drawing in Manifold Spatial DBMS
Export Drawing - DXF

DXF or .dxf is a format used by AutoCAD. Manifold exports to DXF format files suitable for use with AutoCAD 14 and higher versions. Drawings and labels components may be exported to DXF.

To export a drawing to DXF format:

1. Open the drawing in a drawing window.
2. Choose File - Export - Drawing from the main menu.
3. In the Export dialog choose DXF Files in the Save as type box and specify a filename to use. Press Save.
4. Check the Synthesize Z data box and choose a field to be used for Z data if Z data is to be exported into the drawing.
5. Press OK.

Controls

**Synthesize Z data**  
When checked, the field selected in the combo box will be used to generate Z data that will be exported into the drawing. Not checked by default.

Exporting a DXF with elevation data will embed elevation tags into the resulting polyline entities.

Exporting Labels to DXF

Label components may be exported to DXF as follows:

- Labels bound to lines are exported as though they are bound to the centroids of each line.
- Line breaks within multi-line labels are removed.
- Label rotation angles are preserved.
- Exported labels will follow their display order (from lowest to highest).

Exporting Maps to DXF

Map components may also be exported to DXF. Any image and surface layers in the map will be omitted during export to DXF.

Tech Tip

When exporting a drawing or map to a DXF file Manifold will adjust component names by replacing characters that are not supported by AutoCAD with spaces.
Export Drawing - E00

E00 or .e00 format is an ESRI format used for spatial interchange with ArcInfo and similar products. Export to .e00 format is available only with Manifold Enterprise Edition licenses. If you have not purchased Enterprise Edition, you will not be able to export to .e00 format.

To export a drawing to E00 format:

1. Open the drawing in a drawing window.
2. Choose File - Export - Drawing from the main menu.
3. In the Export dialog choose E00 Files in the Save as type box and specify a filename to use. Press Save.
4. Check the fields that are to be exported. Buttons are provided for Select All, Select None and Select Inverse to rapidly change which fields are to be exported for drawings containing many fields.
5. Press OK.

Controls

Select All - Check all fields. All data fields in the drawing will be exported.

Select None - Uncheck all fields. No data fields will be exported. Tech tip: Use this to uncheck all fields before checking those desired.

Select Inverse - Uncheck all previously checked fields and check all previously unchecked fields. This is a fast way to check only a few columns out of many: uncheck those desired and then push Select Inverse.

Technical Note

All Manifold editions can import .e00 files containing both vector and raster data, along with coordinate system (projection) information.

Manifold Enterprise Edition can export vector and raster data in .e00 format as well. Exporting vector data generates the following data sections: ARC (double-precision), CNT, PAL, PRJ, IFO (with .AAT, .BND, .PAT and other subsections as necessary). Exporting vector data to .e00 creates .e00 files analogous to .e00 files of "coverages" exported by ESRI products.

.e00 is an undocumented format that apparently has been changed by ESRI over time. Although Manifold's .e00 import and export capabilities will obviously attempt to follow any changes introduced by ESRI, there is no guarantee that what works today with current versions of ESRI products will also work with future ESRI products that did not exist at the time this Manifold release was created. For best compatibility with the widest range of ESRI products that can read or write .e00, make sure to upgrade to the latest Manifold release and service pack available.

See Also

Enterprise Edition
Export Drawing - KML, KMZ

KML is an XML-based format originally used for “annotations” in Google Earth displays. KMZ is exactly the same format compressed using “zip” compression with a three-letter file name extension of .kmz. KML can be used for showing points or other vector objects from a GIS format drawing as well as for images. Manifold includes the ability to export a drawing or image as a Google Earth KML or KMZ file. It's a quick and convenient way of publishing your GIS data for use with Google Earth or other applications that can use KML, such as NASA World Wind, Microsoft Virtual Earth applications and others.

Manifold can also import KML and KMZ. See the Import Drawing - KML, KMZ topic for additional import information.

KML and KMZ support Latitude / Longitude projection using the WGS 84 Auto datum only. Drawings to be exported to KML or KMZ can be in any projection and will be re-projected on the fly into Latitude / Longitude during export; however, to avoid a potentially slow re-projection process it is wise to explicitly re-project drawings into Latitude / Longitude using the WGS 84 Auto datum before export. This not only avoids a slow export, it also provides an opportunity to catch errors before export if the re-projection results in a strange display as might happen if the initial projection was not correctly assigned, say, after importing projected data from some format that does not store projection information.

Unlike drawings, images must be already in Latitude / Longitude projection using the WGS 84 Auto datum to be exported to KML or KMZ.

KML format as produced for Google Earth is a simplified format that records only the geographic locations and shapes of objects and up to two fields per object, a Name field and a Description field. It does not allow storage of more sophisticated database information. If Manifold fields other than text fields are used for the Name and Description, Manifold will capture the data as text for storage into the KML file. Missing or blank field values will be ignored when a drawing or map is exported to KML.

Drawings exported to KML or KMZ can contain points, lines and areas. Areas will be rendered using partial opacity while points and lines will be rendered at full opacity. To use transparency with points and lines, export a map which has the drawing as a layer with layer opacity set to the desired value; however, areas are always rendered using partial opacity with that partial opacity being modified by whatever layer opacity setting has been used in a map. That is, areas can be made even more transparent but can not be made fully opaque.

To export a drawing to KML format:

1. Open the drawing in a drawing window.
2. Choose File - Export - Drawing from the main menu.
3. In the Export dialog choose KML Files in the Save as type box and specify a filename to use. Press Save.
4. In the Export KML File dialog if desired specify the columns in the drawing's table that will be used for Name and Description fields in the KML file.
5. If you would like to extrude points, lines or areas into 3D shapes, specify a column in the drawing's table that will be used for the Height to give the height in meters. By default this is meters above sea level. Check the Relative heights box to extrude the object to that height in meters above the terrain elevation.
6. Press OK.

Exporting Maps

To export all drawings in a map to KML or KMZ, follow the above procedure with the map. Individual map layers are exported as separate folders in the KML. Note that all drawings in the map should have columns with the same names if such column names are specified for Name, Description or Height.

KML or KMZ exported from maps will respect layer transparency (opacity). That is, if a layer has had partial transparency set using the Opacity setting for a layer, upon export to KML the objects displayed in Google Earth will also be partially transparent. This can be used to create outstanding effects.

When a map is exported, folders in the resultant KML for drawings that have more than ten objects will be closed by default. This prevents the folder display in Google Earth from growing overly large by default.

Creating KMZ Files
The Manifold KML exporter will create .kmz files if the extension for the filename given is .kmz. KMZ format is simply a "zipped" KML file.

**About the WGS 84 Auto Datum**

This is a special datum based on the standard WGS84 datum that has automatic fine adjustment to account for different image servers or applications that use WGS84. When projecting a component for ultimate export to KML or KMZ, use the WGS 84 Auto datum.

**Extruding 3D Shapes**

Specifying a Height column that contains a value for each object in the drawing to use as an altitude will **extrude** the object into a 3D shape of that height in meters above sea level. Checking the Relative heights box (unchecked by default) will convert that height to a relative height above ground level in that location. It is usually a good idea to check the Relative heights box.

The above illustration shows gerrymandered congressional districts as areas that were exported to Google Earth using a Height column that gave values of tens of thousands of meters. The districts are very large, so to have them be obviously higher than the surrounding terrain requires large heights. The areas were colored using the same colors for both area background and also area border line. Google Earth shows the border line in opaque rendering even as the main area is partially opaque. This is a pleasant effect in that it results in nice highlights for the "walls" of the extruded shape.
For the above illustration black color was used for area border color. Choosing a good color combination for area color and area border line color is an important part of the art of making appealing Google Earth displays.

**Coordinate Accuracy**

Exporting a drawing to a KML file preserves at least 8 decimal digits in coordinate values and trims trailing zeros. This typically results in a file that is smaller yet retaining full precision for coordinates.

**Caution**

Export to Google KML and KMZ is an experimental capability due to the constantly evolving and poorly documented nature of Google Earth itself. Google Earth appears to have been designed mainly for visual overview by consumers and not as a rigorous GIS application for use by professionals requiring full precision. Of special importance is that essential documentation on the use of projections and the data composition methods used to prepare raster data for display by Google's mapping services is absolutely lacking.

Therefore, while Manifold export to KML and KMZ makes it easy to create striking visual displays with very little effort, users should not expect that at very detailed zoom levels the objects exported by Manifold will line up perfectly with images generated by Google Earth. The Manifold objects will be placed to absolute precision given the limits of whatever data source has provided those objects, but Google may not always be in agreement with data sources such as USGS, the Census Bureau or US military data sources.

Because Google provides little information on how it cobbles up a particular visual display, it is not possible to diagnose why a particular Google image sometimes does not match commonly accepted, authoritative sources of GIS data that are written to KML or KMZ using the information provided by Google.

Given the relative newness of Google Earth and the focus on consumer viewing, it is possible that it is simply not a matter of concern for the application that some imagery might have shifts of a few meters or even a few dozen meters as compared to GIS data from other sources. After all, since the announced goal of the application is consumer "mash ups" showing things like the locations of one's favorite beer halls it is more important that points of interest may be declared in a simple and easy-to-apply format than that such points appear with accuracy better than a few meters. One suspects that tight integration with the many sources of professional GIS data might be such a low priority matter that Google itself may not at the present time have investigated all details necessary to resolve data compatibility issues as can be done within dedicated GIS applications.

Given Google's often-stated commitment to supporting "open" development by the Google user community we expect Google Earth to evolve and be improved in many ways, including support for closer matches between user-supplied data and Google Earth displays when using KML and KMZ. Should Google Earth or KML and
KMZ format change before the next edition of Manifold becomes available, it is possible that new versions of export to KML and KMZ format might become available as add-ins to Manifold System.

See Also

A Flashy Demo - Web Queries and KML
Exporting KML to Google Earth
Fun with Google Earth
Linked Images from Google Servers
Export Image - KML, KMZ
Import Drawing - KML, KMZ
Export Drawing - MIF

MapInfo .mid/.mif format is widely used for interchange with MapInfo and other GIS applications. The format normally uses two files that have the same base name but with a .mif or .mid extension. The exporter asks for a mif file name and will create a similarly named mid file as well. When we use the term "mif" file in this topic we mean both the .mif file and its .mid companion as well.

Formatting

Mif files can contain color information as well as size parameters for objects. Manifold will read foreground color, background color and size information for objects on import. When exporting to mif format, Manifold will save foreground color, background color and size as well.

Exporting a drawing as a mif sets the outline color of each exported area to the area border foreground color.

Data Types

On import of a MapInfo .mif or .tab file, Manifold will translate MapInfo logical data type columns into Boolean data type. Exporting a Manifold drawing to .mif will export Boolean columns into MapInfo logical data type columns.
Export Drawing - Oracle

Requires Manifold Enterprise Edition or above. This topic provides a typical example of exporting a drawing to a spatial DBMS using Oracle. The same procedure can also be used to export drawings to any spatial DBMS, including DB 2 with Spatial Extender, PostgreSQL/PostGIS, SQL Server 2008, SQL Server 2005 with Manifold Spatial Extender, and just about any DBMS using Manifold's ability to create and use generic spatial indices.

Manifold Enterprise Edition or higher editions can export drawings to Oracle data sources, saving the drawing attributes within the database and drawing geometry as SDO_GEOMETRY as used by Oracle Spatial databases as well as the Locator facility within standard Oracle databases. Export of drawings to Oracle databases is not supported by Manifold editions below Enterprise.

To export a drawing to an Oracle database:

1. Open the drawing in a drawing window.
2. Choose File - Export - Drawing from the main menu.
3. In the Export dialog choose Data Sources () in the Save as type box.
4. Connect to the desired Oracle data source, creating a data source for the Oracle server desired if one has not already been created in the Data Source dialog.
5. Provide the information required in the Export Drawing dialog. Choose Oracle in the Type box to export using Oracle SDO_GEOMETRY and Oracle spatial facilities. Choose Manifold in the Type box to use Manifold-managed spatial facilities. Check the fields that are to be exported and other options as desired.
5. Press OK.

Export Drawing Dialog Controls

<table>
<thead>
<tr>
<th>Data Source</th>
<th>The name of the data source selected in the Data Source dialog.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Either Oracle or Manifold. Choose Oracle in the Type box to export using Oracle SDO_GEOMETRY and Oracle spatial facilities. Choose Manifold in the Type box to use Manifold-managed spatial facilities.</td>
</tr>
<tr>
<td>Name</td>
<td>Name of the new table to create in the server in which this drawing will be stored. By default, Manifold will try to construct a valid Oracle name from the name of the drawing.</td>
</tr>
<tr>
<td>Select All</td>
<td>Check all fields. All data fields in the drawing will be exported.</td>
</tr>
<tr>
<td>Select None</td>
<td>Uncheck all fields. No data fields will be exported. Tech tip: Use this to uncheck all fields before checking those desired.</td>
</tr>
<tr>
<td>Select Inverse</td>
<td>Uncheck all previously checked fields and check all previously unchecked fields. This is a fast way to check only a few columns out of many: uncheck those desired and then push Select Inverse.</td>
</tr>
<tr>
<td>Columns</td>
<td>Check the box for each column (field) that is to be exported with the drawing. By default, Manifold will export all columns except intrinsic columns. Note: The ID column and the Geom(I) column will not appear in the Columns pane because they are always exported.</td>
</tr>
<tr>
<td>Identity</td>
<td>Allows customizing the name of the identity column. OID by default.</td>
</tr>
<tr>
<td>Geometry</td>
<td>Allows customizing the name of the geometry column. Geometry by default.</td>
</tr>
<tr>
<td>Geometry type</td>
<td>Disabled when used with Oracle type, since SDO_GEOMETRY will be used as the geometry type.</td>
</tr>
</tbody>
</table>
### Import and Export

Enabled when Manifold type is used to allow specification of the desired geometry type from a choice of Geometry, Geometry (SHP) or Geometry (WKB). Use Geometry for maximum performance and robustness.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Version column</strong></td>
<td>Allows customizing the name of the version column. Version by default.</td>
</tr>
<tr>
<td><strong>Precision</strong></td>
<td>Allows customizing the value of the location precision parameter. Expressed in whatever native drawing units are used by that drawing.</td>
</tr>
<tr>
<td><strong>Projection</strong></td>
<td>The projection to use when storing the drawing, which is chosen by default from the projections supported by the server. Manifold will use the same projection used by the drawing if it is supported by the server (nearly always possible) or whichever is the most similar projection available on the server. Press the [...] to manually alter the projection chosen by Manifold. Using a projection on the server different than that used by the drawing will require re-projecting the drawing, which can take a significant amount of time for large drawings. Disabled when Manifold type is used since Manifold always stores an exact match when Manifold manages the spatial storage.</td>
</tr>
<tr>
<td><strong>Do not reproject data</strong></td>
<td>Off by default. Check to prevent re-projection of data on export, as might be required to match the projection in use by Manifold to an available Oracle projection. Disabled when Manifold type is used since Manifold always stores an exact match when Manifold manages the spatial storage.</td>
</tr>
<tr>
<td><strong>Create spatial index</strong></td>
<td>Checked by default. Creates an Oracle spatial index for the table containing the drawing, which makes certain Oracle spatial operations faster. Creates a Manifold spatial index when Manifold manages the spatial storage.</td>
</tr>
<tr>
<td><strong>Create sequence and triggers</strong></td>
<td>Checked by default. Creates a sequence object to generate values for the ID column, a trigger to fill values in the ID column for newly inserted records, and another trigger to update values in the Version column for changed records. Disabled when Manifold type is used since Manifold always creates sequences and triggers when Manifold manages the spatial storage.</td>
</tr>
</tbody>
</table>

**Note**

The default name of the identity column used when exporting a drawing to an Oracle database is **OID** to avoid naming collisions with the **ID** column that stores Manifold object IDs in Manifold drawings.

**Exporting Tables with Variable-Length Columns to Oracle**

Oracle tables cannot have more than one variable-length column. Attempting to export a drawing with more than one variable-length column to an Oracle database will fail with a message that lists the variable-length columns and advises converting one or more of these columns to fixed-length types, or unchecking them in the list of exported columns.

**Projections**

Manifold supports all Oracle projections (coordinate systems) although Oracle does not support all Manifold projections. When a component is imported or linked from an Oracle database it will automatically use that same projection within Manifold.

When a component is exported from Manifold to Oracle, Manifold will automatically choose the Oracle projection that is the best match, that is, the most similar to, the Manifold projection in use. Manifold export dialogs will
report the degree of similarity between the Manifold projection and the proposed Oracle coordinate system. If need be, Manifold will re-project the component on the fly into that Oracle projection system as part of the uploading process.

There are two nuances of interest in the case of components using a coordinate system that appears both in Oracle and in Manifold (that is, all Oracle coordinate systems):

- Although Manifold supports all coordinate systems used by Oracle, sometimes the names used for those coordinate systems are different. For example, the Latitude / Longitude projection used in Manifold is known as the TWD97 projection in Oracle when using the WGS 84 datum (the default for Latitude / Longitude).
- Manifold defines coordinate system presets using higher-precision parameters than those used in Oracle so even if the coordinate system is otherwise identical the Manifold settings have to be lowered in precision to match those used by Oracle. Manifold understands such precision-matching requirements when identifying the equivalent projection in Oracle and will automatically re-project data to account for precision differences. If this or any other re-projection is not desired, Manifold provides a Do not project data option (off by default) that may be checked on to prevent any re-projection of data.

**Errors**

If an export fails Manifold will display any error message returned by the Oracle server.

Exporting a drawing with the Create spatial index option turned on will postpone creating a spatial index until after all drawing objects have been uploaded. This is to assure that all uploaded data will stay in the database if index creation fails.

**See Also**

Spatial DBMS
Oracle Spatial Facilities
Example: Storing a Drawing in Oracle
Manifold Spatial DBMS Facilities
Example: Storing a Drawing in Manifold Spatial DBMS
Export Drawing - SDTS

Spatial Data Transfer Standard (SDTS) is a US Federal government format for spatial data interchange. SDTS may be used to export drawings in only a few projections (a limitation of the format): latitude / longitude, UTM, UPS and State Plane. If the drawing to be exported is not in one of these projections it should be re-projected into them before export to SDTS is attempted. Exporting a drawing into SDTS format that is not in one of the supported SDTS projections will convert the drawing into latitude / longitude projection automatically.

To export a drawing to SDTS format:

1. Open the drawing in a drawing window.
2. Choose File - Export - Drawing from the main menu.
3. In the Export dialog choose SDTS Files in the Save as type box and specify a filename to use. Press Save.
4. Check the fields that are to be exported. Buttons provide for Select All, Select None and Select Inverse to rapidly change which fields are to be exported for drawings containing many fields.
5. Optionally, specify the maximum number of coordinates that can make up a single line or area object and press OK.

Controls

- Select All - Check all fields. All data fields in the drawing will be exported.
- Select None - Uncheck all fields. No data fields will be exported. Tech tip: Use this to uncheck all fields before checking those desired.
- Select Inverse - Uncheck all previously checked fields and check all previously unchecked fields. This is a fast way to check only a few columns out of many: uncheck those desired and then push Select Inverse.

Use ring registration method

On by default. One of two allowed registration methods specified in the SDTS specification. Use this for compatibility with older SDTS-reading programs (like MicroDEM) that can only handle ring registration.

Limit line and area objects to …

coordinates

The maximum number of coordinates that can make up a single line or area object before it is split into multiple entities.

Although STDS is a fine format, it does have the limitation that individual objects (lines or areas) cannot be made up of more than approximately 5000 coordinates (also known as "inflection points" or "vertices" in some GIS packages). Some SDTS viewing programs set a maximum limit even lower than 5000 coordinates per object. The Manifold SDTS export dialog allows specification of the maximum number of coordinates allowed per object. Objects consisting of more than the given number of coordinates will be automatically split by Manifold into multiple entities in the saved SDTS file.

Exporting a surface to SDTS format will create a data quality module for compliance with the latest SDTS draft standard.
Export Drawing - SHP, Shapefiles

ESRI's .shp format, also known as "shape format" or "shapefiles," is used with ArcView, a popular GIS package of the early 1990's. Shape format has been openly published by ESRI and is widely used for data interchange in GIS. Shapefiles usually consist of three similar named files with differing extensions: a .shp, .shx and a .dbf file. The .dbf file is a dBase database system format file that is used to store data attributes for the drawing.

When Manifold exports to "shp" format it creates three files: a .dbf, a .shp and a .shx file. These work fine in all applications that can read "shapefiles" correctly. When providing the result of your export to shapefiles to someone else, don't forget to provide all three files and not just the .shp file. Because shapefiles cannot contain a mixture of areas, lines and points, when a drawing that contains a mix of areas, lines and points is exported to shapefile format Manifold must create shapefiles for the areas, shapefiles for the lines and shapefiles for the points.

When exporting drawings containing objects of only one type (only areas or only lines or only points) to shapefiles no postfixes will be appended to the filename. When drawings contain more than one type of object, Manifold will add "a," "l" and "p" postfixes to the filenames to indicate which shapefiles contain areas, lines and points.

Drawings exported as shapefiles should contain at least one data column in the drawing's table to avoid confusing unsophisticated shapefile-reading software. If there are no columns in the drawing's table or if we deselect all fields when exporting, then exporting the drawing as a shapefile will result in "shapefiles" that do not have a .dbf file. This may confuse some third-party software packages that do not understand how to read shapefiles with no .dbf files. Attempting to export a shapefile with no data columns will display a confirmation message warning of this risk that asks users to press Yes to continue.

To export a drawing to .shp format:

1. Open the drawing in a drawing window.
2. Choose File - Export - Drawing from the main menu.
3. In the Export dialog choose SHP Files in the Save as type box and specify a filename to use. Press Save.
4. Check the fields that are to be exported. Buttons provide for Select All, Select None and Select Inverse to rapidly change which fields are to be exported for drawings containing many fields.
5. Press OK.

Controls

- **Select All** - Check all fields. All data fields in the drawing will be exported.
- **Select None** - Uncheck all fields. No data fields will be exported. Tech tip: Use this to uncheck all fields before checking those desired.
- **Select Inverse** - Uncheck all previously checked fields and check all previously unchecked fields. This is a fast way to check only a few columns out of many: uncheck those desired and then push Select Inverse.

Shapefiles as most commonly occur do not save projection information. Recent extensions to ArcView allow the saving of projected data in additional files; however, most shapefiles that are published do not use this extension. Shape format is therefore best used to interchange data from unprojected drawings using degrees as the unit of measure. For maximum compatibility with other GIS applications it is strongly recommended that only drawings in Latitude / Longitude projection be saved to .shp format. To do this, before exporting the drawing use Edit - Change Projection to re-project the drawing into Latitude / Longitude projection if it is in a different projection.

The formal .shp specification includes use of dBase .dbf format. In Microsoft Windows operating systems, most applications use Microsoft drivers to read and write the .dbf part of shapefiles. Windows .dbf drivers normally support standard dBase format, which has a variety of legacy limitations. For example, .dbf files (and thus any shapefiles) should have have "8.3" style names, where the base name is at most eight characters, followed by a .shp, .shx, or .dbf extension. In addition, field names used within a shapefile should not be more than eight alphanumeric characters not beginning with a number.
Because Manifold and most other modern GIS software allows field names to be longer than eight characters and use special characters as well, it is a frequent occurrence that field names used within Manifold are too long or complex for use within shapefiles that are truly compliant with the .dbf standard used in Windows. Such field names should be renamed before export to .shp format is accomplished.

We realize that many users of shape files will rename files to longer than eight characters and also use field names greater than eight characters. This occurs especially often in UNIX environments where programs will at times use .dbf file drivers that are not consistent with dBase specification in that they relax the naming conventions used by dBase. However, writing to longer file names and using longer field names is a violation of the .dbf spec and may well cause failed interchange with other software that takes the spec seriously.

Because so many ESRI applications, both in UNIX and in Windows, do not respect Windows standards for .dbf, Manifold System does not use Microsoft drivers to read / write .dbf files when importing or exporting shapefiles. Instead, a custom set of Manifold drivers are used that accept nonstandard file and field names. In addition, the Manifold drivers will allow writing of long file and field names that violate the dBase standard but which are acceptable to various ESRI applications.

Manifold's shapefile export will preserve data types consistent with .dbf practice whenever possible. Variable-length text data is exported as fixed-length text with 254 characters (since ArcExplorer 2.x and some other programs do not seem to be able to handle memo fields).

**Internationalization**

Exporting a .dbf file (either by exporting a table as a .dbf or by exporting a drawing as a shapefile) will set the .dbf codepage field to the codepage used by the columns in the exported table. If the columns use conflicting codepages or if the codepage is not supported by .dbf format, Manifold will display a warning message.

Importing a .dbf file (either by importing a table from a .dbf or by importing a drawing from a shapefile) sets the codepage of each text column to that used in the .dbf file.

**Exporting Projected Shapefiles**

Because .shp format does not capture projection information it is unwise to export projected drawings into .shp format. However, if for some reason we absolutely must export projected data we should keep in mind the raw nature of data in projected form and the options used to represent locations in projected coordinate systems.

For example, suppose we have a drawing in some metric projection that uses local offsets of 100, 100 and local scales of 10, 10. Suppose we have a point the coordinates of which are 1, 2 in this coordinate system. When exporting this drawing as a .shp, sometimes we may want the coordinate numbers locating the point in the .shp file to be 1, 2 and sometimes 110, 120.

The Manifold .shp exporter does not transform the coordinate numbers in any way, so Manifold will always export 1, 2 for the coordinates of the point. If desired, we can force Manifold to export 110, 120 by first re-projecting the drawing into the coordinate system using local offsets of 0 and local scales of 1.

**Exporting a PRJ File**

To create a PRJ file to accompany the shapefiles for those software packages that can use ESRI PRJ files, use the Edit - Change Projection dialog's toolbar to save the coordinate information into a PRJ file. Name the file using the same name as the other shapefiles with the extension .prj.

**Troubleshooting**

It is tempting to use Manifold as a format converter, for example, to import from MapInfo .mif format and to save as .shp. However, since MapInfo like Manifold also allows longer file and field names than does .shp format it is very easy to import files that already contain field names or use file names incompatible with export to .shp. Such names must be renamed to fit into the limitations of .shp format before export.

If the results of .shp export from Manifold cannot be read by your .shp reading application, check the following:
Are all filenames in the "8+3" form? If not simplify the file names. Your application may not be able to handle long file names.

Are all field names no more than eight, simple alphanumeric (letters and numbers only) and not beginning with a number?

Have you used only simple database types (integers, fixed length text, etc.) for data attributes? Your application might not be able to recognize more sophisticated data types such as dates.

See Also

Edit - Change Projection

Export Image

Exporting Images

Export images from Manifold by right-clicking on the image in the project pane and choosing export, or by opening the image and then using the File - Export - Image dialog. In addition, images can be exported into databases using spatial DBMS technology can be exported by using the Data Source dialog. See the Example: Storing an Image in Oracle and Example: Storing an Image in Manifold Spatial DBMS topics for a typical example of exporting to a spatial DBMS as can also be done for DB 2 with Spatial Extender, PostgreSQL/PostGIS, SQL Server 2008, SQL Server 2005 with Manifold Spatial Extender, and just about any DBMS using Manifold's ability to provide generic spatial DBMS facilities.

When using the File - Export - Image dialog the Files of type choice acts as a filter for what is displayed in the browse pane in the Export dialog. Simple file types are exported directly from the Export dialog by choosing the desired type in the Files of type box. More complex exports into data sources such as databases are exported by choosing Data Sources () in the Files of type box and then using the Data Source dialog.

Images may be exported to various formats. In some cases, such as AI, EMF, PS and PDF the image export occurs not from an image window but rather from a print layout where the layout is exported as an image to the target format.

To export an image:

1. Open the image to be exported in an image window.
2. Choose File - Export - Image
3. Choose the desired format in the Save as type box.
4. Use the resulting dialog to choose export options and to export.

Manifold can export images to a variety of popular formats.

- **AI** Adobe Illustrator format. Exported from a print layout.
- **BMP** Windows bitmap format. Provides no compression at all.
- **ECW / JPEG2000** ERMapper ECW format using wavelet compression or in JPEG 2000 format depending upon the file name extension. Images of any size can be exported.
- **EMF** Extended Windows Meta File format. Often used for interchange with modern graphics packages like Adobe Illustrator.
- **GIF** Older format used for smaller web images. Manifold does not directly support saves to .gif due to Unisys patent threats. Third parties have published free extensions to Manifold that install the ability to write .gif as if it were a built-in capability. If you have such an extension installed you can write .gif.

   For saving images for the web, use .png instead, which can be read by all web browsers and is a much more efficient format than .gif. See the note on .gif usage.
Import and Export

**JPEG**
Classic and very efficient compressed format. Often used for web images that must preserve good appearance. Options include:
- **Quality** - Set to Low, Medium, High or Maximum to trade off image size for quality. The higher the quality the lower the compression and thus the larger the image.
- **Method** - Set to Baseline, Baseline Optimized or Progressive. When Progressive is chosen, set the number of Scans desire.

**KML, KMZ**
Google Earth XML-based KML or KMZ format.

**PDF**
Adobe Portable Document Format, an amusing spin on the word “portable” to mean exclusive use within Adobe’s world view. In all fairness, PDF is a rather handy way of publishing a print layout to an image file for redistribution. Exported from a print layout.

**PNG**
Simple graphics format that is a popular replacement for .gif in websites. PNG is cool. See the Just Say No to GIFs essay.

**PS**
PostScript PS format, a generic format used by many Adobe products. Exported from a print layout.

**TGA**
Targa TGA format. A Blast from the Past still in occasional use.

**TIFF**
Tagged image file format. When the Export projection info checkbox is checked will automatically write as a GeoTIFF format file. GeoTIFF files still use the .tif three letter extension as do ordinary .tif files and, in fact, are simply ordinary .tif files but with a few extra tags in them to store projection information.

**DB2 Data Sources**
Connect to IBM DB2 equipped with the DB2 Spatial Extender using IBM’s native spatial connection technology. A spatial DBMS connection accessed through the Data Source dialog. This option requires Enterprise Edition or above.

**Oracle Data Sources**
Connect to an Oracle data sources using the Oracle Call Interface (OCI), the native Oracle interface for exchanging data. Exchanging images with Oracle data sources via OCI automatically maps images into Oracle GeoRaster form (if GeoRasters are supported by the Oracle product in use). A spatial DBMS connection accessed through the Data Source dialog. This option requires Enterprise Edition or above.

**PostgreSQL Data Sources**
Connect to PostgreSQL equipped with the PostGIS spatial extender using native PostgreSQL connection technology. Connections made using this data source will have any password required masked so that later usage will not inadvertently expose the password. A spatial DBMS connection accessed through the Data Source dialog. This option requires Enterprise Edition or above.

**SQL Server Data Sources**
Connect to Microsoft SQL Server 2008 spatial DBMS facilities using native SQL Server 2008 spatial connection technology. A spatial DBMS connection accessed through the Data Source dialog. This option requires Enterprise Edition or above.

**Important:**
Using a vendor’s native spatial DBMS connection technology (like those above) provides better performance and allows using features, such as built-in spatial DBMS facilities, not exposed through generic database interfaces such as ODBC, OLE DB and ADO .NET. At the present writing, only Oracle Spatial provides a built-in GeoRaster type. Other spatial DBMS products are supported using...
JPEG images are frequently used as images in web pages. Several different quality settings can be selected depending on the tradeoff desired between image size and image quality.

JPEG images can be arranged so that they load as one byte stream (Baseline) or as a sequence of updates (Progressive). If you've ever seen an image appear in a web page where initially a simplified view of the image appears and then more detail fills in via horizontal swaths you were likely looking at a Progressive JPEG image. The number of scans sets forth how many iterative updates are used to build the image. Progressive images are larger than Baseline images, but when served over a slow Internet connection can be more tolerable because the viewer sees at least a schematic view of the image right away.

The Baseline Optimized choice provides a higher quality JPEG at lower size than normal; however, some older web browsers are not able to display optimized JPEG images.

Image Storage within Databases

In addition to storage within Oracle Spatial databases, Manifold has a variety of means of storing images within external databases. See the Virtual Tables for Images and Surfaces topic, the Linked Images topic and the Queries and Images or Surfaces topic.

Manifold can automatically save images into databases using either vendor-supplied spatial DBMS technology as in the case of Oracle Spatial GeoRasters or with almost any DBMS using Manifold-managed spatial DBMS technology. Storing images in a fast DBMS is often the fastest way to handle large images.

Transparent Pixels and Export

GIF, PNG and TIFF support transparent pixels. GIF images do not have an alpha channel, so when exporting GIF files Manifold will render RGBa images over the background color specified in Tools - Options.

XML Files Created upon Export and Used on Import

Not all formats to which Manifold can export have the ability to correctly save projection information. As a safety measure to ensure that projection information is never lost, Manifold always writes an accompanying .xml file when exporting drawings, images, labels, maps or surface components to a file. The accompanying .xml file contains coordinate system information and some other metadata. The XML file is created for all formats, even those that correctly save coordinate system information.

When Manifold imports drawings, images, labels or surfaces, the system will check for an accompanying .xml file that might have been written by Manifold. If such an .xml file exists, Manifold will read it and use the information it contains to load a correct coordinate system.

See the XML Accessory File Format topic for details on the internal organization of this file.

A Note on GIF Usage

It is a great idea to avoid using .gif format. Use .png instead: it's faster, more efficient and it is not subject to moronic patent threats from Unisys. See Just Say No to GIFs for a quick history on the threat to .gif usage from Unisys.

See Also

Tools - Make Image
Export Image - AI, EMF, PS and PDF
Export Image - GeoTIFF
Export Image - Oracle
Export Image - PNG
Export Image - AI, EMF, PS and PDF

Manifold offers four quasi-vector graphics arts / imaging formats to use when exporting print layouts as "images."

- **AI** A graphics arts format read by Adobe Illustrator and other applications.
- **EMF (GDI)** Enhanced Meta File format, the 32 bit version of WMF (Windows Meta File) format. This option writes EMF using classic Windows GDI (Graphics Device Interface). Use the GDI version as a possible workaround only if the GDI+ version does not provide satisfactory results.
- **EMF (GDI+)** EMF written using GDI+. Microsoft’s GDI+ was developed for Windows XP and is installed by Manifold on all Windows systems if not already installed. The GDI+ version should be used by default, as GDI+ includes numerous improvements over GDI.
- **PDF** Portable Document Format, read by Adobe Acrobat Reader and other applications.
- **PS** Export to PostScript compatible files

**Options**

All four formats use a similar dialog to code the output image and include these options:

- **Pixel resolution** Used to save images and surfaces that are in the layout. Resolution must be greater than 1 DPI and less than 7200 DPI. Default is screen resolution, normally 72 DPI.
- **Vector resolution** Used to save drawings and labels as well as layout elements such as text or horizontal and vertical lines. Resolution must be greater than 1 DPI and less than 7200 DPI. Default is 300 DPI. If saving for subsequent work in some other program before printing choose a Vector resolution setting that is the same as the resolution of the printer that will be used. If printing to a 600 DPI printer, for example, use 600 DPI.
- **Ignore Styles** Export the image using default point, line, area and label styles.

In addition, the **Export PS File** dialog includes two more options used when exporting to PS:

- **Pixel Encoding** Choose Flat, ASCII HEX or Run-length. The default choice is Flat, which tends to produce smaller files. The other two choices are supplied as possible workarounds if encountering problems with other PS reading software.
- **Use explicit image mask for invisible pixels** Switches between two different methods for handling invisible pixels allowed in PS format. When checked (the default) each pixel includes an extra byte that specifies if the pixel is transparent or opaque. When not checked, invisible pixels are set to an infrequently used color (R:G:B = 0:0:1) that is then designated transparent. Unchecking the box will reduce file size for files containing large images, but may cause problems for inept PS reading software. AI format does not support invisible pixels.

In addition, the **Export PDF File** dialog includes options used when exporting to PDF:

- **Compress output** Use internal compression algorithms allowed in PDF format to reduce the size of the resulting PDF file.
Enabled by default.

**Use transparency**  Export using transparency, so that any partially transparent components will be rendered in the .pdf with partial transparency.

**Ignore styles**  Export using default styles for points, lines and areas.

The **Ignore styles** checkbox is an advanced option that forces Manifold to use default styles for exported objects. This can be useful when exporting for later use in vector-style graphics editors such as Illustrator because the default styles produce a relatively small number of simple vector objects per Manifold entity. In contrast, more complex styles available in Manifold can produce five to ten relatively complex vector objects per point. If the export is destined for use in Illustrator and the Illustrator operator prefers to work with simple objects check this box.

**Exporting Multipage Layouts**

Multipage layouts may be exported to **PDF** and **PS**, but not to **EMF**. However, a single page of a multipage layout may be exported to **EMF**. When a multipage layout is exported to **EMF**, the Export EMF File dialog will include a **Page** option that allows specification of which page in the multipage layout is to be exported.

When exporting a multipage layout to **PDF** or **PS**, a **Pages** option (called a **page filter**) appears that controls which pages from the layout appear in the exported image. Enter a series of page numbers or page ranges separated by commas. White space is ignored and reverse ranges are interpreted correctly. For example, Enter **1, 2, 5-8** or **1, 2, 8-5** and get the same results. See the Multipage Layouts topic for an example of using page filters.

**About GDI and GDI+**

Manifold uses Windows system routines wherever possible for maximum performance and continuing quality as Windows updates are released. The Graphics Device Interface (GDI) is a Windows subsystem used for a variety of utility tasks involving device-independent graphics, including export to EMF format. The original GDI has now been replaced by GDI+, which was developed with Windows XP and has been released by Microsoft in DLLs that developers can install on earlier Windows systems. Manifold automatically installs GDI+ on Windows systems other than XP.

The GDI+ version of the EMF exporter should always be used by default. If problems occur when using the resulting files in other programs, try using the GDI version to see if reverting to the earlier Windows technology solves the problem (as it might with some programs that were tested or developed using GDI and not GDI+).

**Compatibility**

Although the formats in this topic are often thought to be "universal," device independent formats the usage of these formats is no guarantee that the visual appearance of what one program writes will be recovered when the file is read by a different program. For example, **.emf** exports using ordinary **GDI** do support opacity for images and surfaces but not for drawings.

All four formats in this topic may be thought of as programming languages as much as they are formats. They operate not only by providing data but also by providing instructions on how a document is to be dynamically constructed based upon that data. Whenever a program reads one of these programs it must synthesize the desired document using the data and instructions contained in the file. The program must do its best to synthesize the desired document even if its capabilities or the facilities of its host system (such as fonts that might be available) may be less able than those in the system that wrote the file.

The full range of possibilities for describing a document within these four systems can be very complex and extensive and few programs (if any) that read or write these formats have implemented all of the capabilities that are theoretically possible. As a result, there can be considerable variation in how different programs interpret the instructions for synthesizing a document contained in these formats. Small differences in capability or interpretation can result in documents that are visibly different.
Export Image - ECW / JPEG2000

ECW and JPEG 2000 are two different formats that are handled by the same File - Export - Image choice in Manifold. Which format is written depends upon the three letter extension used for the file name. Manifold can export either ECW or JPEG 2000 images without limitations on size other than, of course, the capacity of the machine in use and the patience of the user.

ECW or JPEG 2000 format is strongly recommended for very large images. Manifold can link an image from an ECW or JPEG 2000 file with great speed. Images linked from these formats that are many gigabytes in size will load and display in just a few seconds.

ECW and JPEG 2000 are two different formats even though they are handled by the same exporter within Manifold. ECW is strongly preferred because it can store projection information (see notes below).

To export an image as an ECW file:

1. Open the image to be exported in an image window.
2. Choose File - Export - Image
3. Choose ECW / JPEG2000 in the Save as type box.
4. Enter a file name for the file ending in .ecw and press Save.
5. In the Export ECW / JPEG2000 File dialog enter a Compression ratio and press OK.

To export an image as a JPEG 2000 file:

1. Open the image to be exported in an image window.
2. Choose File - Export - Image
3. Choose ECW / JPEG2000 in the Save as type box.
4. Enter a file name for the file ending in .j2k, .j2c, .jp2, .jpc, .jpf, or .jpx and press Save.
5. In the Export ECW / JPEG2000 File dialog enter a Compression ratio and press OK.

Important: We must explicitly specify the three letter extension, either .ecw or one of the allowed three letter .j-- extensions, to tell the exporter whether we want an ECW file or a JPEG 2000 file. Do not forget to specify this extension (it is not something Windows users are accustomed to have to do) as forgetting to specify the extension will result, unpredictably, in the exporter choosing on its own which format to use.

Compression Ratio

Set the compression ratio desired, such as 1:10 or 1:20 or even greater in the export dialog and the ECW / JPEG 2000 routines will do their best to compress the image to the degree specified. ECW export supports all projections available within ECW files.

Note that wavelet compression can provide remarkable compression without apparent degradation of the image; however, both ECW and JPEG2000 are "lossy" formats and are not appropriate for certain types of remote sensing data where not even a single pixel can be change. These formats are best for visual data where appearance to the human eye is the main criterion.

Projections

ECW format can save projection information although the list of projections supported by ECW does not include all projections supported by Manifold. (Manifold has grown to support so many projections that virtually no other software supports such a long list of projections).

Although JPEG 2000 in theory can save projection information, in actual practice for the reasons given below it should not be treated as a projection-aware format. When Manifold writes a JPEG 2000 file it will also create an accompanying .xml file giving full projection information. In the special case of JPEG 2000 files written by Manifold that still are accompanied by the .xml accessory file, Manifold will be able to write and read projection information perfectly with JPEG 2000 files.
Import and Export

However, that capability of storing projection information will not extend to JPEG 2000 files written by other applications which we try to import into Manifold, nor is it likely that other applications will be able to read projection information from JPEG 2000 files written by Manifold. See the discussion below for reasons why.

**Good News and Bad News for Open Standards**

The ECW and JPEG 2000 code within Manifold System provides a living example of both the benefits and the curse of cooperation between companies using open source and open standards. First, the good news:

The export code to export to both ECW and to JPEG 2000 used within Manifold is open source code provided by ERM (Earth Resource Mapping, the makers of ERMapper) to the GIS community. ERM invented ECW and developed it into an effective format for storing compressed images. ECW is not only elegant from an academic, computer science perspective, it is also supremely practical in that it incorporates "real world" requirements such as the need to store projection information in a perfect, unambiguous form. Years of focused improvement under the guidance of expert remote sensing applications programmers have also given ECW extraordinary performance.

To help establish ECW as a universal format, ERM had the great presence of mind to provide source code to ECW to other GIS and remote sensing companies. In addition, ERM implemented the new JPEG 2000 "open" standard for compression within a common software development kit (SDK) for which freely available source code was openly provided. Having both formats supported by the same SDK with freely available source code removed any tension vendors might have as to which standard to support. It also showed that ERM had the confidence in ECW to allow the industry at large to make a choice even if the use of JPEG 2000 became free and effortless.

The ECW / JPEG 2000 SDK is now used throughout the GIS and remote sensing communities and is cooperatively maintained for use in UNIX, Linux and Windows by industry professionals with ERM managing the standard and taking the lead in UNIX and Manifold and others assisting with Windows notes. The objective is to have a single code base for all platforms to assure that the resulting ECW or JPEG 2000 files are fully interoperable.

Overall, the effort has been very successful, a shining example of how mutual support using access to source code can propel a commercial standard like ECW into universal acceptance by commercial vendors. About the only compromise required is that a side effect of using such shared code within Manifold for this exporter is the requirement to explicitly specify the three letter extension, either .ecw or one of the allowed three letter j-- extensions, to indicate whether an ECW or a JPEG 2000 is required. This small matter aside, the release of source code for the ECW / JPEG2000 SDK by ERM has resulted in superb performance and quality for ECW.

And now, the bad news of "open" standards:

Given ERMapper’s extensive experience in image compression, ERM had the experience to develop practical, effective and high quality JPEG 2000 code and has produced what is indisputably a best-of-breed JPEG 2000 implementation. The problem is that the JPEG 2000 standard itself has been weakened by too many "design by committee" compromises instead of a solid focus on practical software implementation and use. Those compromises may have been well-intentioned but they have resulted in a format that does not work as well for GIS use as ECW.

In particular, the JPEG 2000 standard attempts to incorporate too many different approaches in the matter of projections information storage and so succeeds in being darn near useless in this key area. It allows several different methods to store projection information in a JPEG 2000 file.

With different ways of storing projection information JPEG 2000 opens itself up to a Tower of Babel phenomenon where no two packages are likely to use the same method in exactly the same way. Even in the case of packages that attempt to support multiple methods or the same method the intrinsic ambiguity of the methods usually chosen for JPEG 2000 make it unlikely that projection information interchange will be accomplished in all cases. We can see why by looking at the best of the methods allowed in JPEG 2000, EPSG codes.

The most useful of projection storage methods within JPEG 2000 is the use of EPSG codes. These are also used by OGC (the inventors of famously incompetent "standards" like GML) for things like OGC WMS / WFS and are used by Manifold for the Manifold IMS implementation of OGC WMS / WFS as well.

For all the merits of EPSG, the available EPSG codes do not cover a sufficient number of projections. A further problem is that EPSG codes are not defined in a single, unique set and so different applications can end up using code sets that are not identical. As a result, even the use of EPSG codes imposes great limitations on what can be inside a JPEG 2000 and how that information can be interchanged. As the industry moves to greater standardization and a single set of EPSG codes this situation should be improved.
The JPEG 2000 inability to make decisions in a unifying way can be seen from the very nature of JPEG 2000 filenames. A standard which cannot make up its mind which three letter extension to use and instead compromises by allowing six different names for essentially the same thing (.j2k, .j2c, .jp2, .jpc, .jpf, or .jpx) obviously from the very beginning has a serious problem with squishy thinking. If the standard cannot even standardize its own name there is little hope it will do better on more contentious issues, such as methods for encoding projections.

Note that in no way do we suggest that JPEG 2000 is not well intentioned. It's actually a great project done for all the right reasons with the best of intentions. But as anyone who has ever participated in one of the committees that earnestly come up with such things can attest, it is good intentions that pave the road to you-know-where. It could well be that the press of evolution and the indisputable talent of the many people who work on JPEG 2000 will take a hand in guiding JPEG 2000 into more useful form. Perhaps there will emerge a JPEG 2007 or JPEG 2010 that one day will equal ECW in effectiveness. Until then, GIS professionals with a job to do will reach first for ECW.

Because source code for ECW is now freely available within the GIS and remote sensing vendor community, users and vendors can trust that ECW will continue to serve as the de facto standard for "real world" GIS and remote sensing applications. For the reasons set forth above, manifold.net strongly encourages Manifold users to employ ECW in preference to JPEG 2000.

Tech Tip

JPEG 2000 may be squishy about filenames but that doesn't mean you have to be. If for some reason you choose to save files into JPEG 2000 format, consider always using the .j2k three letter extension. There's no particular reason for this advice except that .j2k looks like a more memorable mnemonic for JPEG 2000 than, say, "jpc" or the others. Although the original suggestion for JPEG 2000 was to use .jp2 as an extension, it appears that .j2k is now slightly more popular than the other choices. Within the Manifold community, we may as well voluntarily "standardize" on .j2k.

See Also

Compressed Images
Export Image - GeoTIFF

To export a georegistered image to GeoTIFF format, use the File - Image - Export dialog using TIFF in the Save as type box.

TIFF is Adobe’s copyrighted “tagged image file format” that provides a surprisingly rich set of extensions. It is often used to save simple images without any georegistration. However, it also provides for added information through the inclusion of extra tags. Software that does not understand how to use extra tags simply ignores them.

The GeoTIFF organization has defined a set of standard tags that can be used to embed projection / georegistration information within TIFF files. When a TIFF file includes such tags it is said to be a GeoTIFF file.

When Manifold reads an existing GeoTIFF it automatically reads the projection tags and will import the GeoTIFF with correct coordinate system info. When Manifold writes a TIFF, by default, Manifold will export projection information tags into a TIFF so that it is automatically made a GeoTIFF. There is no separate File - Export - Image setting to write a "GeoTIFF" because writing a TIFF always creates a GeoTIFF if the Export projection info box has been checked in the export dialog.

To export an image as a GeoTIFF file:

1. Georegister the image if it is not already georegistered.
2. Open the image to be exported in an image window.
3. Choose File - Export - Image
4. Choose TIFF files (*.tif, *.tiff) in the Save as type box.
5. Enter a file name for the file and press Save.
6. Choose Intel (IBM PC) in the byte order box.
7. For maximum interchangeability, choose None in the compression box.
8. Check the Export projection info box and Press OK.

Options

- **Byte Order**
  - Intel (IBM PC) - Use for all PC clones (AMD, Intel, etc.).
  - Motorola (Mac) - Use for Apple Mac products.

- **Compression**
  - None - Best choice for interchange. Not all software products can read compressed TIFF files.
  - JPEG, PackBits, LZW, ZIP - Various compression methods in common use with GeoTIFF files. Not all software packages can read these.

- **Export projection info**
  - Checked by default. Export projection info tags into the TIFF. This makes it a GeoTIFF. Don’t uncheck unless you want to create an ordinary TIFF that contains no projection tags.
TIFF export does not currently support JPEG compression for palette images. This is a bit odd, since JPEG export itself supports palette images. To create a JPEG compressed GeoTIFF from a palette image, first convert the image to RGB or grayscale for export. See the Image Types topic.

**XML Files Created upon Export and Used on Import**

Not all formats to which Manifold can export have the ability to correctly save projection information. As a safety measure to ensure that projection information is never lost, Manifold always writes an accompanying .xml file when exporting drawings, images, labels, maps or surface components to a file. The accompanying .xml file contains coordinate system information and some other metadata. The XML file is created for all formats, even those that correctly save coordinate system information.

When Manifold imports drawings, images, labels or surfaces, the system will check for an accompanying .xml file that might have been written by Manifold. If such an .xml file exists, Manifold will read it and use the information it contains to load a correct coordinate system.

See the XML Accessory File Format topic for details on the internal organization of this file.

**Tech Tip**

Keep inexpert users away from your GeoTIFFs because such users cannot seem to resist killing those GeoTIFFs by opening them with some brainless consumer graphics editing package. See the comments in the Import Image - TIF [Various types] topic.

If you do have a GeoTIFF killed by an inexpert user and that GeoTIFF was created by Manifold, if you still have the auxiliary XML file created by Manifold the situation can be saved. Place the GeoTIFF and its accompanying XML file in the same folder so that when you open the former GeoTIFF the projection information will be restored. Save the GeoTIFF again and the projection information will be re-embedded into what will be once again a GeoTIFF when it is saved. The inclination of inexpert users to kill GeoTIFFs was one of the main reasons Manifold introduced an auxiliary XML file to save projection information, just in case.
Export Image - KML, KMZ

KML is an XML-based format used by Google Earth for "annotations" in Google Earth displays. KML can be used for showing points or other vector objects from a GIS format drawing as well as for images. Manifold includes the ability to export a drawing or an image as a Google Earth KML file provided the component is in Latitude / Longitude projection using the WGS 84 Auto datum. It's a quick and convenient way of publishing your GIS data for use with Google Earth.

To export an image to KMZ format:

1. Open the image. Images must be in Latitude / Longitude using the WGS 84 Auto datum for export to KML or KMZ, so re-project the image into Latitude / Longitude using the WGS 84 Auto datum if necessary.
2. Choose File - Export - Image from the main menu.
3. In the Export dialog choose KMZ Files in the Save as type box and specify a filename to use. Press Save.

Creating KMZ or KML Files

The Manifold KML / KMZ exporter will create .kmz files if the extension for the filename given ends in .kmz and it will create .kml files if the filename given ends in .kml. KMZ format is simply a "zipped" version of KML.

Use KMZ when exporting images because then the image itself will be embedded within the .kmz file that is created. If KML is used, then Manifold will create both a separate image file and also a .kml file. These must be kept together for display in Google Earth. Given the possibility that the image file might not always accompany the .kml file that uses that image, it is wise to always use .kmz in such circumstances to make sure the image always accompanies the relevant KML.

About the WGS 84 Auto Datum

This is a special datum based on the standard WGS84 datum that has automatic fine adjustment to account for different image servers or applications that use WGS84. When projection a component for ultimate export to KML or KMZ, use the WGS 84 Auto datum.

Caution

Export to Google KML and KMZ is an experimental capability due to the constantly evolving and poorly documented nature of Google Earth itself. Google Earth appears to have been designed mainly for visual overview by consumers and not as a rigorous GIS application for use by professionals requiring full precision. Of special importance is that essential documentation on the use of projections and the data composition methods used to prepare raster data for display by Google's mapping services is absolutely lacking.

Therefore, while Manifold export to KML and KMZ makes it easy to create striking visual displays with very little effort, users should not expect that at very detailed zoom levels the objects exported by Manifold will line up perfectly with images generated by Google Earth. The Manifold objects will be placed to absolute precision given the limits of whatever data source has provided those objects, but Google may not always be in agreement with data sources such as USGS, the Census Bureau or US military data sources.

Because Google provides little information on how it cobbles up a particular visual display, it is not possible to diagnose why a particular Google image sometimes does not match commonly accepted, authoritative sources of GIS data that are written to KML or KMZ using the information provided by Google.

Given the relative newness of Google Earth and the focus on consumer viewing, it is possible that it is simply not a matter of concern for the application that some imagery might have shifts of a few meters or even a few dozen meters as compared to GIS data from other sources. After all, since the announced goal of the application is consumer "mash ups" showing things like the locations of one's favorite beer halls it is more important that points of interest may be declared in a simple and easy-to-apply format than it that such points appear with accuracy better than a few meters. One suspects that tight integration with the many sources of professional GIS data might be such a low priority matter that Google itself may not at the present time have investigated all details necessary to resolve data compatibility issues as can be done within dedicated GIS applications.

Given Google's often-stated commitment to supporting "open" development by the Google user community we expect Google Earth to evolve and be improved in many ways, including support for closer matches between user-supplied data and Google Earth displays when using KML and KMZ. Should Google Earth or KML and
KMZ format change before the next edition of Manifold becomes available, it is possible that new versions of export to KML and KMZ format might become available as add-ins to Manifold System.

Tech Tip

The general Google definitions for KML and KMZ are broader than what might be the capabilities of a particular Google product that uses these formats. Users intending to employ these formats with Google products should take time to read Google's technical documentation in detail to understand how to do so correctly for each particular Google technology.

See Also

A Flashy Demo - Web Queries and KML
Exporting KML to Google Earth
Fun with Google Earth
Linked Images from Google Servers
Export Drawing - KML, KMZ
Import Drawing - KML, KMZ
Export Image - Oracle

Requires Manifold Enterprise Edition or above. This topic provides a typical example of exporting an image to a spatial DBMS using Oracle. The same procedure can also be used to export images to any spatial DBMS using Manifold's ability to save images using generic spatial DBMS capability.

Manifold Enterprise Edition or higher editions can export images into Oracle's GeoRaster database storage technology as found within Oracle Spatial and other GeoRaster capable Oracle products. This is a very fast storage medium that is excellent for very large images and can be utilized by any application that is Oracle GeoRaster compatible. Oracle Spatial scales very well: whether stored images are a few hundred megabytes, a few gigabytes or even a terabyte the database response is fast.

The Oracle server in use must have GeoRaster capability. Note that some Oracle products, such as Oracle Express Edition, have Locator spatial capability but **do not** provide GeoRaster capability. If the Oracle server in use does not have GeoRaster capability, Manifold can still save images into the database using Manifold spatial facilities. Manifold image storage is also very fast, although it cannot scale across Oracle clusters the way GeoRaster can.

To export an image into an Oracle Spatial database we must have access to that database and have write permissions.

**To export an image to an Oracle database:**

1. Open the image in an image window.
2. Choose *File - Export - Image* from the main menu.
3. In the *Export* dialog choose *Data Sources ()* in the *Save as type* box.
4. Connect to the desired Oracle data source, creating a data source for the Oracle server desired if one has not already been created in the Data Source dialog.
5. Provide the information required in the *Export Image* dialog. Choose *Oracle* in the *Type* box to export using Oracle GeoRaster if available. Choose *Manifold* in the *Type* box to use Manifold image storage within the database. Check the other options as desired.
6. Press OK.
Type  Either Oracle or Manifold. Choose Oracle in the Type box to export using Oracle GeoRaster and Oracle spatial facilities. Choose Manifold in the Type box to use Manifold-managed image storage.

Name  Name of the new table to create in the server in which this image will be stored. By default, Manifold will try to construct a valid Oracle name from the name of the image.

Projection  The projection to use when storing the image, which is chosen by default from the projections supported by the server. Manifold will use the same projection used by the image if it is supported by the server (nearly always possible) or whichever is the most similar projection available on the server. Press the [...] to manually alter the projection chosen by Manifold. Using a projection on the server different than that used by the image will require re-projecting the image, which can take a significant amount of time for large images. Disabled when Manifold type is used since Manifold always stores an exact match when Manifold manages the spatial storage.

Do not reproject data  Off by default. Check to prevent re-projection of data on export, as might be required to match the projection in use by Manifold to an available Oracle projection. Disabled when Manifold type is used since Manifold always stores an exact match when Manifold manages the spatial storage.

Create pyramids  Create intermediate level images to enable much faster zooming and panning. Always leave this checked unless you are a maximum Oracle GeoRaster expert and have some specialized reason not to create pyramids.

Split data into tiles  On by default. If checked, will split image data into tiles of the specified size instead of sending image data in one chunk. Disabled when Manifold type is used since Manifold always uses tiles.

Tile size  Size of image tile to use in pixels. The value estimated by Manifold is usually the best choice.

Compression  Disabled within Oracle since GeoRaster uses Oracle's own technology. Enabled when Manifold type is used and effectively always on by default since compression is faster than not and is lossless.

Projections

Manifold supports all Oracle projections (coordinate systems) although Oracle does not support all Manifold projections. When a component is imported or linked from an Oracle database it will automatically use that same projection within Manifold.

When a component is exported from Manifold to Oracle, Manifold will automatically choose the Oracle projection that is the best match, that is, the most similar to, the Manifold projection in use. Manifold export dialogs will report the degree of similarity between the Manifold projection and the proposed Oracle coordinate system. If need be, Manifold will re-project the component on the fly into that Oracle projection system as part of the uploading process.

There are two nuances of interest in the case of components using a coordinate system that appears both in Oracle and in Manifold (that is, all Oracle coordinate systems):

- Although Manifold supports all coordinate systems used by Oracle, sometimes the names used for those coordinate systems are different. For example, the Latitude / Longitude projection used in Manifold is known as the TWD97 projection in Oracle when using the WGS 84 datum (the default for Latitude / Longitude).
Manifold defines coordinate system presets using higher-precision parameters than those used in Oracle so even if the coordinate system is otherwise identical the Manifold settings have to be lowered in precision to match those used by Oracle. Manifold understands such precision-matching requirements when identifying the equivalent projection in Oracle and will automatically re-project data to account for precision differences. If this or any other re-projection is not desired, Manifold provides a Do not project data option (off by default) that may be checked on to prevent any re-projection of data.

See Also

Oracle Spatial Facilities
Example: Storing an Image in Oracle
Export Image - PNG

To export an image to PNG format, use the **File - Image - Export** dialog using PNG in the **Save as type** box. After a filename is provided and we press OK a secondary **Export PNG File** dialog will be raised with options.

Options

- **Interlaced**: If checked, will interlace the resultant image so that it can download progressively when used in web publishing. Using interlaced images will slightly expand the image size, but it is psychologically easier on visitors to web sites when large images are served because the visitor can see the large images being downloaded progressively.

- **Embed description info**: If checked, write text properties into the .png image using the component name as the **Title** property, the component's description text property as the **Description** property and the **Software** property set to the "Manifold System..." identification string displayed in the **Help - About** dialog.

The **Embed description info** option is a useful way of embedding copyright slogans: enter the copyright text or whatever other text you want embedded into the image into the component's **Description** property and it will be embedded into the .png image that is exported. Since very few people know that the .png format can contain embedded text information, virtually no one who copies the resultant .png image will realize that you have, in effect, "watermarked" it with embedded .png text properties.

**Notes**

.png is fine format for web publishing and for graphics arts, but it does not preserve georeferencing and projection information. Use GeoTIFF to preserve projection info. It is the ethical duty of all webmasters to do their best to replace .gif format with .png.

**See Also**

Just Say Not to GiFs

Export Layout

Exporting Layouts

Print Layouts are used to arrange jobs for printing. They may also be exported to file formats. To save the layout to a file, we choose **File - Export - Image**. The **File - Export - Image** menu pathway is used for all formats, including those, such as .pdf, which are quasi-vector in nature.

To export a layout to a file:

1. Create the layout and open it.
2. Choose **File - Export - Image**.
3. In the **Export** dialog, browse to the folder in which the file is to be saved, choose a filename and in the **Save as type** box choose the file format to be used. Press **Save**.
4. Formats such as .emf, .pdf, and .ps will raise another **Export ... File** dialog. Choose options desired and press **OK**.

**File Types**

- **AI**: A graphics arts format read by Adobe Illustrator and other applications.
- **EMF (GDI)**: Enhanced Meta File format, the 32 bit version of WMF
Import and Export

(Windows Meta File) format. This option writes EMF using classic Windows GDI (Graphics Device Interface). Use the GDI version as a possible workaround only if the GDI+ version does not provide satisfactory results.

**EMF (GDI+)**
EMF written using GDI+. Microsoft’s GDI+ was developed for Windows XP and is installed by Manifold on all Windows systems if not already installed. The GDI+ version should be used by default, as GDI+ includes numerous improvements over GDI.

**PDF**
Portable Document Format, read by Adobe Acrobat Reader and other applications.

**PS**
Export to PostScript compatible files

Compatibility

Although the .ai, .emf, .pdf and .ps formats are often thought to be "universal," device independent formats the usage of these formats is no guarantee that the visual appearance of what one program writes will be recovered when the file is read by a different program. For example, .emf exports using ordinary GDI support opacity for images and surfaces but not for drawings.

The .ai, .emf, .pdf and .ps formats in this topic may be thought of as programming languages as much as they are formats. They operate not only by providing data but also by providing instructions on how a document is to be dynamically constructed based upon that data. Whenever a program reads one of these programs it must synthesize the desired document using the data and instructions contained in the file. The program must do its best to synthesize the desired document even if its capabilities or the facilities of its host system (such as fonts that might be available) may be less able than those in the system that wrote the file.

The full range of possibilities for describing a document within these formats can be very complex and extensive and few programs (if any) that read or write these formats have implemented all of the capabilities that are theoretically possible. As a result, there can be considerable variation in how different programs interpret the instructions for synthesizing a document contained in these formats. Small differences in capability or interpretation can result in documents that are visibly different.

See Also

- Export Layout - AI
- Export Layout - EMF
- Export Layout - PDF
- Export Layout - PS
Export Layout - Ai

Export a print Layout to an Ai file. Ai is a graphics arts format read by Adobe Illustrator and other applications.

To export a layout to an Ai file:

1. Create the layout and open it.
2. Choose File - Export - Image.
3. In the Export dialog, browse to the folder in which the file is to be saved, choose a filename and in the Save as type box choose Ai file format. Press Save.
4. In the subsequent dialog, choose options desired and press OK.

Options for Ai export

Pixel resolution Used to save images and surfaces that are in the layout. Resolution must be greater than 1 DPI and less than 7200 DPI. Default is 72 DPI.

Vector resolution Used to save drawings and labels as well as layout elements such as text or horizontal and vertical lines. Resolution must be greater than 1 DPI and less than 7200 DPI. Default is 300 DPI. If saving for subsequent work in some other program before printing choose a Vector resolution setting that is the same as the resolution of the printer that will be used. If printing to a 600 DPI printer, for example, use 600 DPI.

Pages Export only specified pages from a multipage layout.

Convert text to vector shapes When turned on (default), text elements are converted to vector shapes, as happens when outlining fonts in Illustrator. When off, text elements will be exported as text using the defined font whenever possible but will use the system font (Helvetica) instead of the original font if preservation of fonts is not possible. Using vector shapes results in a larger file but one that is faithful to the appearance of fonts originally used in all circumstances. Not converting text to vector shapes preserves the font for possible editing in Illustrator using Illustrator's text tools.

Ignore Styles Export the image using default point, line, area and label styles.

The Ignore styles checkbox is an advanced option that forces Manifold to use default styles for exported objects. This can be useful when exporting for later use in vector-style graphics editors such as Illustrator because the default styles produce a relatively small number of simple vector objects per Manifold entity. In contrast, more complex styles available in Manifold can produce five to ten relatively complex vector objects per point. If the person who will be operating Illustrator prefers to work with simple objects check this box.

Default Resolutions for Export

Screen resolution is normally 96 DPI on most monitors. The default resolution for layout exports is set to 72 DPI because at 72 DPI one point is one pixel. For example, at 72 DPI an 8 pt Tahoma font will be 8 pixels wide.

See Also

Exporting Layouts

Layouts
Export Layout - EMF

Export a print Layout to an EMF file. EMF is Enhanced Meta File format, the 32 bit version of WMF (Windows Meta File) format. It may be written in Manifold using either GDI or GDI+ engine.

To export a layout to an EMF file:

1. Create the layout and open it.
2. Choose File - Export - Image.
3. In the Export dialog, browse to the folder in which the file is to be saved, choose a filename and in the Save as type box choose either the EMF (GDI+) file format (recommended) or EMF (GDI). Press Save.
4. In the subsequent dialog, choose options desired and press OK.

Options for EMF export

- **Pixel resolution**: Used to save images and surfaces that are in the layout. Resolution must be greater than 1 DPI and less than 7200 DPI. Default is 72 DPI.
- **Vector resolution**: Used to save drawings and labels as well as layout elements such as text or horizontal and vertical lines. Resolution must be greater than 1 DPI and less than 7200 DPI. Default is 300 DPI. If saving for subsequent work in some other program before printing choose a Vector resolution setting that is the same as the resolution of the printer that will be used. If printing to a 600 DPI printer, for example, use 600 DPI.
- **Page**: Export only the specified page from a multipage layout.
- **Ignore Styles**: Export the image using default point, line, area and label styles.

The Ignore styles checkbox is an advanced option that forces Manifold to use default styles for exported objects. This can be useful when exporting for later use in vector-style graphics editors such as Illustrator because the default styles produce a relatively small number of simple vector objects per Manifold entity. In contrast, more complex styles available in Manifold can produce five to ten relatively complex vector objects per point. If the person operating Illustrator prefers to work with simple objects check this box.

Default Resolutions for Export

Screen resolution is normally 96 DPI on most monitors. The default resolution for layout exports is set to 72 DPI because at 72 DPI one point is one pixel. For example, at 72 DPI an 8 pt Tahoma font will be 8 pixels wide.

About GDI and GDI+

Manifold uses Windows system routines wherever possible for maximum performance and continuing quality as Windows updates are released. The Graphics Device Interface (GDI) is a Windows subsystem used for a variety of utility tasks involving device-independent graphics, including export to EMF format. The original GDI has now been replaced by GDI+, which was developed with Windows XP and has been released by Microsoft in DLLs that developers can install on earlier Windows systems. Manifold automatically installs GDI+ on Windows systems other than XP.

The GDI+ version of the EMF exporter should always be used by default. If problems occur when using the resulting files in other programs, try using the GDI version to see if reverting to the earlier Windows technology solves the problem (as it might with some programs that were tested or developed using GDI and not GDI+).

See Also

Exporting Layouts

Layouts
Export Layout - PDF

Export a print Layout to a PDF file. PDF is Portable Document Format, read by Adobe Acrobat Reader and other applications.

To export a layout to a PDF file:

1. Create the layout and open it.
2. Choose File - Export - Image.
3. In the Export dialog, browse to the folder in which the file is to be saved, choose a filename and in the Save as type box choose PDF file format. Press Save.
4. In the subsequent dialog, choose options desired and press OK.

Options for PDF export

- **Pixel resolution**: Used to save images and surfaces that are in the layout. Resolution must be greater than 1 DPI and less than 7200 DPI. Default is 72 DPI.
- **Vector resolution**: Used to save drawings and labels as well as layout elements such as text or horizontal and vertical lines. Resolution must be greater than 1 DPI and less than 7200 DPI. Default is 300 DPI. If saving for subsequent work in some other program before printing choose a Vector resolution setting that is the same as the resolution of the printer that will be used. If printing to a 600 DPI printer, for example, use 600 DPI.
- **Pages**: Export only specified pages from a multipage layout.
- **Compress output**: Use a compression algorithm to save space.
- **Convert text to vector shapes**: When turned on (default), text elements are converted to vector shapes. When off, text elements will be exported as text but will use the system font (Helvetica) instead of the original font. Using vector shapes results in a larger file but one that is faithful to the appearance of fonts originally used.
- **Use layers**: Use .pdf layers.
- **Use transparency**: Enable transparency to correctly render transparent colors.
- **Ignore Styles**: Export the image using default point, line, area and label styles.

The Ignore styles checkbox is an advanced option that forces Manifold to use default styles for exported objects. This can be useful when exporting for later use in vector-style graphics editors such as Illustrator because the default styles produce a relatively small number of simple vector objects per Manifold entity. In contrast, more complex styles available in Manifold can produce five to ten relatively complex vector objects per point. If the person operating Illustrator prefers to work with simple objects check this box.

Default Resolutions for Export

Screen resolution is normally 96 DPI on most monitors. The default resolution for layout exports is set to 72 DPI because at 72 DPI one point is one pixel. For example, at 72 DPI an 8 pt Tahoma font will be 8 pixels wide.

Tech Tip: Printing a Layout to PDF

There are two ways to create a PDF file from a layout: exporting a layout to a PDF as described within this topic or "printing" the layout via File - Print using a PDF virtual printer driver, which mimics a printer to capture any printing activity as a PDF.
Printing a layout to a virtual printer driver that produces PDF feeds the layout through an ordinary GDI / GDI+
rendering pipeline, which issues "dumb" printing commands that are then captured by the driver. Exporting a
layout directly as a PDF feeds the layout through the Manifold PDF rendering engine, which optimizes printing
commands. In general, the results of exporting a layout as a PDF will almost always be better than the results of
printing the same layout to a virtual printer driver.

Note also that printing a layout to a via **File - Print** exposes you to the risk of whatever that PDF printer driver
vendor has in mind. For example, it could easily produce different results depending on the installed version of
Acrobat if that is what is used for a PDF printer driver. In contrast, exporting a layout to a PDF via **File - Export -
Image** will be consistently the same.

**See Also**

- Exporting Layouts
- Layouts
- Printing
Export Layout - PS

Export a print Layout to a PS file. PS is a PostScript compatible format.

To export a layout to a PS file:

1. Create the layout and open it.
2. Choose File - Export - Image.
3. In the Export dialog, browse to the folder in which the file is to be saved, choose a filename and in the Save as type box choose PS file format. Press Save.
4. In the subsequent dialog, choose options desired and press OK.

Options for PS export

- **Pixel resolution**: Used to save images and surfaces that are in the layout. Resolution must be greater than 1 DPI and less than 7200 DPI. Default is 72 DPI.
- **Vector resolution**: Used to save drawings and labels as well as layout elements such as text or horizontal and vertical lines. Resolution must be greater than 1 DPI and less than 7200 DPI. Default is 300 DPI. If saving for subsequent work in some other program before printing choose a Vector resolution setting that is the same as the resolution of the printer that will be used. If printing to a 600 DPI printer, for example, use 600 DPI.
- **Pages**: Export only specified pages from a multipage layout.
- **Pixel Encoding**: Choose Flat, ASCII HEX or Run-length. The default choice is Flat, which tends to produce smaller files. The other two choices are supplied as possible workarounds if encountering problems with other PS reading software.
- **Use explicit image mask for invisible pixels**: Switches between two different methods for handling invisible pixels allowed in PS format. When checked (the default) each pixel includes an extra byte that specifies if the pixel is transparent or opaque. When not checked, invisible pixels are set to an infrequently used color (R:G:B = 0:0:1) that is then designated transparent. Unchecking the box will reduce file size for files containing large images, but may cause problems for inept PS reading software.
- **Ignore Styles**: Export the image using default point, line, area and label styles.

The Ignore styles checkbox is an advanced option that forces Manifold to use default styles for exported objects. This can be useful when exporting for later use in vector-style graphics editors such as Illustrator because the default styles produce a relatively small number of simple vector objects per Manifold entity. In contrast, more complex styles available in Manifold can produce five to ten relatively complex vector objects per point. If the person operating Illustrator prefers to work with simple objects check this box.

Default Resolutions for Export

Screen resolution is normally 96 DPI on most monitors. The default resolution for layout exports is set to 72 DPI because at 72 DPI one point is one pixel. For example, at 72 DPI an 8 pt Tahoma font will be 8 pixels wide.

See Also

Exporting Layouts

Layouts
Export Surface

Exporting Surfaces

Export surfaces from Manifold by right-clicking on the surface in the project pane and choosing export, or by opening the surface and then using the File - Export - Surface dialog. In addition, surfaces can be exported into databases using spatial DBMS technology can be exported by using the Data Source dialog. Surfaces are exported into spatial DBMS as if they were images. See the Example: Storing an Image in Oracle and Example: Storing an Image in Manifold Spatial DBMS topics for a typical example of exporting to a spatial DBMS as can also be done for DB 2 with Spatial Extender, PostgreSQL/PostGIS, SQL Server 2008, SQL Server 2005 with Manifold Spatial Extender, and just about any DBMS using Manifold's ability to provide generic spatial DBMS facilities.

When using the File - Export - Surface dialog the Files of type choice acts as a filter for what is displayed in the browse pane in the Export dialog. Simple file types are exported directly from the Export dialog by choosing the desired type in the Files of type box. More complex exports into data sources such as databases are exported by choosing Data Sources () in the Files of type box and then using the Data Source dialog.

Surfaces may be exported several different formats. If Enterprise Edition is installed, a surface may be exported to ESRI .e00 format as well. Formats supported include:

- **BIL**: ESRI BIL surfaces/terrain elevation
- **ESRI E00**: Surfaces in ESRI .e00 format. [Requires Manifold System Enterprise Edition]
- **ESRI ASCII Grid**: ESRI grid / surface files using ASCII encoding.
- **ESRI Float Grid**: ESRI grid / surface files using binary encoding.
- **KML, KMZ**: Google Earth XML-based KML or KMZ format. Similar to image export: see the Export Image - KML, KMZ topic.
- **Raw Binary Files**: Export the surface to a generic, raw binary format. Creates a data file with the pixels and an XML file with metadata, which includes the coordinate system, dimensions, and type of the surface.
- **SDTS**: Surfaces in SDTS format.
- **Surfer 6 GRD**: Surfer version 6 GRD format
- **Surfer 7 GRD**: Surfer version 7 GRD format
- **Surfer ASCII GRD**: Surfer ASCII GRD format
- **XYZ**: Comma-delimited ASCII files containing X, Y and Z coordinates.
- **DB2 Data Sources**: Connect to IBM DB2 equipped with the DB2 Spatial Extender using IBM's native spatial connection technology. A spatial DBMS connection accessed through the Data Source dialog. This option requires Enterprise Edition or above.
- **Oracle Data Sources**: Connect to an Oracle data sources using the Oracle Call Interface (OCI), the native Oracle interface for exchanging data. Exchanging images with Oracle data sources via OCI automatically maps images into Oracle GeoRaster form (if GeoRasters are supported by the Oracle product in use). A spatial DBMS connection accessed through the Data Source dialog. This option requires Enterprise Edition or above.
- **PostgreSQL Data Sources**: Connect to PostgreSQL equipped with the PostGIS spatial extender using native PostgreSQL connection technology. Connections made using this data source will have any password required masked so that later usage will not inadvertently expose the password. A spatial DBMS connection accessed through the Data Source dialog. This option requires Enterprise Edition or above.
- **SQL Server Data Sources**: Connect to Microsoft SQL Server 2008 spatial DBMS facilities using native SQL Server 2008 spatial connection technology. A spatial DBMS connection accessed through the Data Source dialog. This option requires
Import and Export

Enterprise Edition or above.

**Important:** Using a vendor's native spatial DBMS connection technology (like those above) provides better performance and allows using features, such as built-in spatial DBMS facilities, not exposed through generic database interfaces such as ODBC, OLE DB and ADO.NET. At the present writing, only Oracle Spatial provides a built-in GeoRaster type. Other spatial DBMS products are supported using Manifold-managed image and surface storage.

To export a surface to ESRI .bil format:

1. Open the surface to be exported in a surface window.
2. Choose **File - Export - Surface**
3. Choose **BIL Files (*.bil, *.hdr)** in the **Save as type** box.
4. Enter a file name for the file and press **Save**.

The procedure for exporting a surface to SDTS or .e00 is similar.

When exporting a surface to SDTS, the **Export SDTS Data Set** dialog will be raised with one option, a **Use ring registration method** checkbox. On by default, this checkbox selects one of two registration methods allowed by the SDTS specification. Use this for compatibility with older SDTS-reading programs (like MicroDEM) that can only handle ring registration. Exporting a surface to SDTS format will create a data quality module for compliance with the latest SDTS draft standard.

When exporting to .bil format, in addition to the .bil file Manifold also creates .blw (world file), .hdr (header file), .prj (projection file) and .stx (statistic file) files.

**Exporting to Raw Binary**

The .xml file created when exporting to raw binary will be the name of the exported file with an .xml extension. To learn what fields are included, export a sample surface (such as the Montana Mountain example surface included on the Manifold CD) and then open the .xml file using Notepad.

**Importing from Raw Binary**

If a raw binary surface file is accompanied by an .xml file giving metadata, Manifold will automatically use it to pre-load the **Import Raw Binary File** dialog. See the Import Surface - Raw Binary topic.

**XML Files Created upon Export and Used on Import**

Not all formats to which Manifold can export have the ability to correctly save projection information. As a safety measure to ensure that projection information is never lost, Manifold always writes an accompanying .xml file when exporting drawings, images, labels, maps or surface components to a file. The accompanying .xml file contains coordinate system information and some other metadata. The XML file is created for all formats, even those that correctly save coordinate system information.

When Manifold imports drawings, images, labels or surfaces, the system will check for an accompanying .xml file that might have been written by Manifold. If such an .xml file exists, Manifold will read it and use the information it contains to load a correct coordinate system.

See the XML Accessory File Format topic for details on the internal organization of this file.

**Exporting a GSR File**

To create a GSR file to accompany files written for Golden Software applications like Surfer, use the Edit - Change Projection dialog's toolbar to save the coordinate information into a GSR file. Name the file using the same name as the other files with the extension .gsr.
See Also

Edit - Change Projection

Export Table

Exporting Tables

Export tables from Manifold by right-clicking on the table in the project pane and choosing export, or by opening the table and then using the File - Export - Table dialog. In addition, tables can be exported into databases by using the Data Source dialog.

When using the File - Export - Table dialog the Files of type choice acts as a filter for what is displayed in the browse pane in the Export dialog. Simple file types are exported directly from the Export dialog by choosing Data Sources () in the Files of type box and then using the Data Source dialog.

Any table in Manifold can be exported to common database file formats. Tables can also be exported to ADO.NET, ODBC and OLE DB data sources as well as to Oracle databases using OCI.

<table>
<thead>
<tr>
<th>ADO .NET Data Sources</th>
<th>ADO .NET data sources. Exported using the Data Source dialog.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSV</td>
<td>Comma separated text files. Use this choice and specify the full filename in the File name box complete with extension to save to .asc, .csv, .tab or .txt format. Options allow writing an end comma for each record and quotes around all field values (not just for text fields).</td>
</tr>
<tr>
<td>DB</td>
<td>Paradox .db format.</td>
</tr>
<tr>
<td>DBF</td>
<td>dBase and FoxPro .dbf format.</td>
</tr>
<tr>
<td>HTML</td>
<td>Write as a table within an .htm (HTML) format file, ready for publishing on the web.</td>
</tr>
<tr>
<td>MDB</td>
<td>Microsoft Access .mdb file format.</td>
</tr>
<tr>
<td>ODBC Data Sources</td>
<td>ODBC data sources. Exported using the Data Source dialog.</td>
</tr>
<tr>
<td>OLE DB Data Sources</td>
<td>OLE DB data sources. Exported using the Data Source dialog.</td>
</tr>
<tr>
<td>WKx</td>
<td>Lotus WK tables</td>
</tr>
<tr>
<td>XLS</td>
<td>Microsoft Excel .xls spreadsheet file format.</td>
</tr>
<tr>
<td>DB2 Data Sources</td>
<td>Connect to IBM DB2 equipped with the DB2 Spatial Extender using IBM's native spatial connection technology. A spatial DBMS connection accessed through the Data Source dialog. This option requires Enterprise Edition or above.</td>
</tr>
<tr>
<td>Oracle Data Sources</td>
<td>Connect to an Oracle data sources using the Oracle Call Interface (OCI), the native Oracle interface for exchanging data. A spatial DBMS connection accessed through the Data Source dialog. This option requires Enterprise Edition or above.</td>
</tr>
<tr>
<td>PostgreSQL Data Sources</td>
<td>Connect to PostgreSQL equipped with the PostGIS spatial extender using native PostgreSQL connection technology. Connections made using this data source will have any password required masked so that later usage will not inadvertently expose the password. A spatial DBMS connection accessed through the Data Source dialog. This option requires Enterprise Edition or above.</td>
</tr>
</tbody>
</table>
Import and Export

above.

**SQL Server Data Sources**
Connect to Microsoft SQL Server 2008 spatial DBMS facilities using native SQL Server 2008 spatial connection technology. A spatial DBMS connection accessed through the Data Source dialog. This option requires Enterprise Edition or above.

**Important:** Using a vendor's native spatial DBMS connection technology (like those above) provides better performance and allows using features, such as built-in spatial DBMS facilities, not exposed through generic database interfaces such as ODBC, OLE DB and ADO .NET. For example, exchanging data with Oracle data sources via OCI automatically maps geometry columns into Oracle SDO_GEOMETRY data and will likewise map geometry columns into the native geometry types used by DB2, PostgreSQL and SQL Server 2008.

Export dialogs for **DBF, DB, MDB, WK** and **XLS** include a **Subtype** choice box allowing choice of specific variations within those formats. This is used, for example, to select Access 2000 or Access 97 versions of .mdb format. Export dialog controls include:

**Subtype** Choose format subtype for compatibility with older applications.

- **Select All** - Check all fields. All data fields in the table will be exported.
- **Select None** - Uncheck all fields. No data fields will be exported. Tech tip: Use this to uncheck all fields before checking those desired.
- **Select Inverse** - Uncheck all previously checked fields and check all previously unchecked fields. This is a fast way to check only a few columns out of many: uncheck those desired and then push **Select Inverse**.

**Fields pane** Check the fields to be exported.

**Note:** Exporting a table to a data source that supports NULL values makes intrinsic, rank and script columns (that is, Active columns) NULL-able. That is, the table in the external data source may contain NULL values. In general, exporting a table to an external data source makes all exported columns except the identity column (ID) NULL-able.

**Formatting in Exported Tables**

Tables export to the target format using whatever native field types map best to those in use within Manifold. In general, it is not possible to preserve formatting information such as the number of significant digits to show after a decimal point.

Export to text formats such as .csv or HTML is the exception. Manifold will always export data to these formats as strings formatted using whatever column formats have been applied to the table. This includes use of extra characters such as, for example, the percent character, " %" when it is used as a column formatting style to show a percentage field.

Manifold supports the usual conventions when exporting tables to text files. For example, the first line of the file should contain the field names to be used for the data in the subsequent lines.

**Exporting Tables with Variable-Length Columns to Oracle**

Oracle tables cannot have more than one variable-length column. Attempting to export a drawing with more than one variable-length column to an Oracle database will fail with a message that lists the variable-length columns and advises converting one or more of these columns to fixed-length types, or unchecking them in the list of exported columns.
Important Note when Using 64-bit Manifold Editions

Due to a lack of required Microsoft facilities in 64-bit Windows systems, Manifold in 64-bit mode cannot export, import, export or link to DB, HTML, MDB, XLS or WKx format files. This includes no access to the MDB parts of Manifold MFD and MapInfo TAB imports. The workaround for importing or exporting such files is to launch Manifold in 32-bit mode by using the Manifold System (32-bit) shortcut, perform the export from or import into a .map project file and then re-launch Manifold in 64-bit mode using the Manifold System (64-bit) shortcut.

Linking is more complex: the data must be in some format usable in 64-bit mode within 64-bit Windows systems, such as a SQL Server database. Alternatively, the data can be kept within a Manifold .map project file and linked using the Manifold ODBC driver.
Export Table - ADO .NET / ODBC / OLE DB

ADO .NET, ODBC and OLE DB are database technologies for connecting to a wide variety of database sources. We can connect to databases using these database connection technologies by using the Data Source dialog, creating a new data source in the dialog if necessary.

Exporting a table to an ADO.NET, ODBC, OLE DB or native Oracle data source allows selecting which columns we would like to export.

When exporting a table to an ODBC data source, Manifold tries to preserve, if possible, the signed / unsigned variations of integer types when matching them to the types supported by the data source.

See Also

The Data Source Dialog
Export Table - CSV

CSV refers to a family of ASCII (plain text) formats that save tabular data in the form of one line per record with the values for each field in that record separated by a comma, hence the name Comma Separated Value (CSV) format.

There is considerable variation in how different programs work with "CSV" format. Some programs like to see a comma at the end of each line and others do not. Most programs expect to see only string field values enclosed in quotes, but some like to see all values enclosed in quotation marks whether they are strings or not. Manifold's CSV exporter provides options to deal with both situations.

To export a table to .csv format:

1. Open the table in a table window.
2. Choose File - Export - Table from the main menu.
3. In the Export dialog choose CSV Files in the Save as type box and specify a filename to use. Press Save.
4. Check the fields that are to be exported. Buttons provide for Select All, Select None and Select Inverse to rapidly change which fields are to be exported for tables containing many fields.
5. Check the ending comma and quotes boxes as desired.

Controls

Select All - Check all fields. All data fields in the table will be exported.
Select None - Uncheck all fields. No data fields will be exported. Tech tip: Use this to uncheck all fields before checking those desired.
Select Inverse - Uncheck all previously checked fields and check all previously unchecked fields. This is a fast way to check only a few columns out of many: uncheck those desired and then push Select Inverse.

Fields pane

Write ending comma
Place a comma at the end of each line. Off by default.

Write quotes around non-text fields
Enclose all values within quotation marks even if they are not text fields. If not checked, only text fields will be surrounded by quotation marks. Off by default.

When exporting data to CSV format one loses fine control over how numeric fields are represented. Values end up being either text fields or numbers. A better choice for data interchange in the Microsoft world is to export to Microsoft's .mdb format whenever possible.

When Text Fields Contain Commas

Although the "comma separated values" format is intended to use commas as delimiting characters to separate field values, at times people would like to use different characters. Normally this desire is prompted by text fields that contain commas as part of the data.

Manifold's CSV importer can read "csv" files that use characters other than commas as delimiters but Manifold's CSV exporter only exports using commas (otherwise, it would be somewhat inaccurate to refer to the resulting file as a "comma separated" file). Suppose we do have text fields that contain commas in our table. What can we do? The best approach would be to use .mdb or some other modern format, but if we must export the data into a text file format like CSV there is a workaround:

The usual technique is to open the table and use Edit - Replace to replace all commas with an otherwise unused character, such as the ^ character. Save using the Export Table - CSV command. Open the resulting .csv file with Notepad or some other text editor and replace the commas with the desired delimiter character of choice.
such as a pipe | character or some otherwise unused character. Finally, search and replace all ^ characters with commas. The result will be a "comma separated value" file that uses pipe characters instead of commas.

**See Also**

Import Table - CSV
Export Table - DB

DB refers to a family of formats originally used with the Borland Paradox database management system, with slightly different formats for different versions of Paradox.

To export a table to .db format:

1. Open the table in a table window.
2. Choose File - Export - Table from the main menu.
3. In the Export dialog choose DB Files in the Save as type box and specify a filename to use. Press Save.
4. Check the fields that are to be exported. Buttons provide for Select All, Select None and Select Inverse to rapidly change which fields are to be exported for tables containing many fields.

Controls

Subtype  Choose the Paradox version .db file that is to be created.

- Select All - Check all fields. All data fields in the table will be exported.
- Select None - Uncheck all fields. No data fields will be exported. Tech tip: Use this to uncheck all fields before checking those desired.
- Select Inverse - Uncheck all previously checked fields and check all previously unchecked fields. This is a fast way to check only a few columns out of many: uncheck those desired and then push Select Inverse.

Important Note when Using 64-bit Manifold Editions

Due to a lack of required Microsoft facilities in 64-bit Windows systems, Manifold in 64-bit mode cannot export, import, export or link to DB, HTML, MDB, XLS or WKx format files. This includes no access to the MDB parts of Manifold MFD and MapInfo TAB imports. The workaround for importing or exporting such files is to launch Manifold in 32-bit mode by using the Manifold System (32-bit) shortcut, perform the export from or import into a .map project file and then re-launch Manifold in 64-bit mode using the Manifold System (64-bit) shortcut. Linking is more complex: the data must be in some format usable in 64-bit mode within 64-bit Windows systems, such as a SQL Server database. Alternatively, the data can be kept within a Manifold .map project file and linked using the Manifold ODBC driver.
Export Table - DBF

DBF refers to a family of formats originally used with the dBase database management system. The format is now used with Microsoft's FoxPro DBMS as well as in many Windows and UNIX/Linux DBMS applications that can save database tables. Unfortunately, there is no universal standard for what constitutes a ".dbf" file. The original dBase format limits field names to eight alphanumeric characters not beginning with a number. It also limits file names to eight alphanumeric characters plus a three-letter extension. More recent ".dbf" drivers will allow nine or eleven character filenames and field names. Some UNIX drivers may impose no limitation on file or field names.

To export a table to .dbf format:

1. Open the table in a table window.
2. Choose File - Export - Table from the main menu.
3. In the Export dialog choose DBF Files in the Save as type box and specify a filename to use. Press Save.
4. Check the fields that are to be exported. Buttons provide for Select All, Select None and Select Inverse to rapidly change which fields are to be exported for tables containing many fields.
5. Check use dBase format for greatest compatibility with older databases; however, if this box is checked and field names or file names are longer than 8 simple alphanumeric characters the export will not succeed.

Controls

Subtype Choose dBase or FoxPro .dbf format for compatibility with older applications.

Select All - Check all fields. All data fields in the table will be exported.

Select None - Uncheck all fields. No data fields will be exported. Tech tip: Use this to uncheck all fields before checking those desired.

Select Inverse - Uncheck all previously checked fields and check all previously unchecked fields. This is a fast way to check only a few columns out of many: uncheck those desired and then push Select Inverse.

Fields pane Check the fields to be exported.

One problem with .dbf formats is that the specific nature of what is allowed or disallowed in the format varies greatly depending on the drivers or applications that are used. In particular, some applications can use longer names for files and fields than can other applications using a ".dbf" format. Manifold will use whatever .dbf drivers are available on the host Windows system, by default installing the FoxPro and .dBase .dbf Microsoft drivers.

For maximum compatibility it is best to use filenames and fieldnames that are eight characters or fewer in length. Use only letters and numbers in the name and do not start a filename or field name with a number. Do not use any special characters such as spaces, parentheses or underscores. Most modern drivers will allow 11 character filenames and field names, as do the default .dbf drivers installed with Manifold. However, it is wise to truncate file and field names to only 8 characters if you plan on exchanging data with other ".dbf" capable applications.

Because of the wide variation in drivers and field and file naming conventions in common use, .dbf is not a good choice for reliable data interchange unless it is limited to the least common denominator of using very restricted field and file names. However, such a primitive approach to naming is very costly because it denies us the ability to use descriptive field and file names that are self-documenting.

A better choice for data interchange in the Microsoft world is to export to Microsoft's .mdb format whenever possible.

Internationalization
Exporting a .dbf file (either by exporting a table as a .dbf or by exporting a drawing as a shapefile) will set the .dbf codepage field to the codepage used by the columns in the exported table. If the columns use conflicting codepages or if the codepage is not supported by .dbf format, Manifold will display a warning message.

Importing a .dbf file (either by importing a table from a .dbf or by importing a drawing from a shapefile) sets the codepage of each text column to that used in the .dbf file.
Export Table - MDB

MDB refers to the .mdb formats used with Microsoft's Access database products. These formats have been made available in the Microsoft "Jet" database engines (the same used within Access) that may be used royalty-free by developers working with Microsoft development products such as Visual Studio. As a result, many thousands of third party applications can read and write .mdb formats.

To export a table to .mdb format:

1. Open the table in a table window.
2. Choose File - Export - Table from the main menu.
3. In the Export dialog choose MDB Files in the Save as type box and specify a filename to use. Press Save.
4. Check the fields that are to be exported. Buttons provide for Select All, Select None and Select Inverse to rapidly change which fields are to be exported for tables containing many fields.
5. Check use Access 97 format for greatest compatibility with other applications.

Controls


Select All - Check all fields. All data fields in the table will be exported.

Select None - Uncheck all fields. No data fields will be exported. Tech tip: Use this to uncheck all fields before checking those desired.

Select Inverse - Uncheck all previously checked fields and check all previously unchecked fields. This is a fast way to check only a few columns out of many: uncheck those desired and then push Select Inverse.

Fields pane  Check the fields to be exported.

Access 97 and Access 2000 .mdb Formats

Until Access 2000, all .mdb capable applications could exchange data without any problems. However, with the introduction of Access 2000 and Jet 4.0, Microsoft changed the .mdb format slightly so that the "new" .mdb cannot be read either by Access 97 or by any applications using Jet versions prior to Jet 4.0. In particular, Manifold System Release 4.50 uses Access 97 format.

Manifold’s export to .mdb will write to the "new" .mdb by default. A checkbox option allows writing to the Access 97 .mdb format if desired. Since Access 2000 and Jet 4.0 based applications can read the older Access 97 / Jet .mdb formats, a good strategy to assure maximum compatibility with all ".mdb" applications is to check the Use Access 97 format box if the files will be used for interchange, exchanged with Manifold System 4.50 users or otherwise widely published.

Important Note when Using 64-bit Manifold Editions

Due to a lack of required Microsoft facilities in 64-bit Windows systems, Manifold in 64-bit mode cannot export, import, export or link to DB, HTML, MDB, XLS or WKx format files. This includes no access to the MDB parts of Manifold MFD and MapInfo TAB imports. The workaround for importing or exporting such files is to launch Manifold in 32-bit mode by using the Manifold System (32-bit) shortcut, perform the export from or import into a .map project file and then re-launch Manifold in 64-bit mode using the Manifold System (64-bit) shortcut.

Linking is more complex: the data must be in some format usable in 64-bit mode within 64-bit Windows systems, such as a SQL Server database. Alternatively, the data can be kept within a Manifold .map project file and linked using the Manifold ODBC driver.

Although Microsoft is slowly moving forward towards 64-bit operation of Office and the JET components used by Access and Excel, at the present writing (2010) this is such a total kludge that while it is possible to get 64-bit
access to Access or Excel it is just easier for most to continue using the 32-bit versions. See comments in the 32-bit and 64-bit Manifold Editions topic.
Export Table - XLS

XLS refers to the .xls formats used with Microsoft's Excel spreadsheet products. Routines to read and write these formats have been made available in the Microsoft "Jet" spreadsheet engines that may be used royalty-free by developers working with Microsoft development products such as Visual Studio. As a result, many thousands of third party applications can read and write .xls formats.

Keep in mind that .xls formats are spreadsheet formats. For general database table exchange it is usually wiser to use a database format like Access .mdb.

To export a table to .xls format:

1. Open the table in a table window.
2. Choose File - Export - Table from the main menu.
3. In the Export dialog choose XLS Files in the Save as type box and specify a filename to use. Press Save.
4. Choose the Excel subtype to use.
5. Check the fields that are to be exported. Buttons provide for Select All, Select None and Select Inverse to rapidly change which fields are to be exported for tables containing many fields.
6. Press OK.

Controls

- **Subtype**: Choose which version of Excel format to use. Newer versions of Excel use slightly different .xls formats than earlier versions.
- **Select All**: Check all fields. All data fields in the table will be exported.
- **Select None**: Uncheck all fields. No data fields will be exported. Tech tip: Use this to uncheck all fields before checking those desired.
- **Select Inverse**: Uncheck all previously checked fields and check all previously unchecked fields. This is a fast way to check only a few columns out of many: uncheck those desired and then push Select Inverse.

Fields pane: Check the fields to be exported.

Extra Sheets

Manifold System uses Microsoft's own OLE DB drivers to write Excel format. This provides perfect compatibility with Microsoft's .xls format since Microsoft's own code is used to write the .xls files. Unfortunately, Microsoft's OLE DB driver has an interesting "feature" in that it automatically creates an extra, unnecessary sheet in the Excel file. There is no way to prevent the OLE DB driver from creating this extra sheet.

If Excel is installed on the system, Manifold is able to take extra steps to get rid of the extra sheet. If Excel is installed, Manifold launches an automation routine that locates and instantiates Excel, opens the newly exported file with Excel, turns off confirmations and then deletes the extra sheet. Expert programmers or system administrators may notice an Excel process launching when exporting to .xls - it is being used to eliminate the extra sheet created by the OLE DB driver.

If Excel is not installed on the system, Manifold will be unable to launch Excel to clean up the extra sheet deposited by the OLE DB driver. In that case, each .xls file created by exporting a table will include an extra sheet.

Important Note when Using 64-bit Manifold Editions
Due to a lack of required Microsoft facilities in 64-bit Windows systems, Manifold in 64-bit mode cannot export, import, export or link to DB, HTML, MDB, XLS or WKx format files. This includes no access to the MDB parts of Manifold MFD and MapInfo TAB imports. The workaround for importing or exporting such files is to launch Manifold in 32-bit mode by using the Manifold System (32-bit) shortcut, perform the export from or import into a .map project file and then re-launch Manifold in 64-bit mode using the Manifold System (64-bit) shortcut. Linking is more complex: the data must be in some format usable in 64-bit mode within 64-bit Windows systems, such as a SQL Server database. Alternatively, the data can be kept within a Manifold .map project file and linked using the Manifold ODBC driver.

Although Microsoft is slowly moving forward towards 64-bit operation of Office and the JET components used by Access and Excel, at the present writing (2010) this is such a total kludge that while it is possible to get 64-bit access to Access or Excel it is just easier for most to continue using the 32-bit versions. See comments in the 32-bit and 64-bit Manifold Editions topic.
Projections

Projections are a very important idea in GIS and mapping. If you are not familiar with projections, please begin by reading the Projections Tutorial topic.

This documentation provides several paths through the general subject of working with projections in Manifold. Some material is repeated in the various topics so that essential material is covered no matter which path the reader takes through the Help system.

- Experts may jump directly to the Projections Quick Reference topic for a terse introduction.

- If you are a new user and are working completely within Manifold System, read this topic and then proceed to the Projecting a Map topic. Very little expertise in the inner workings of projections is required to make simple projections when working within Manifold System.

- To permanently re-project a specific component we open it in a window and apply the Edit - Change Projection command. If the component appears in a map and we would like to re-project it to use the same projection as the map, we can right click its tab in the map and choose Project to Map.

- If you must deal with maps in legacy formats such as ESRI .shp or AutoCAD .dxf, please read the Projections and Legacy Formats topic. Because legacy formats do not normally save information on projection parameters used, importing maps saved in legacy formats can require extensive knowledge of projections and coordinate systems.

- If you are new to projections and coordinate systems in GIS and would like to learn more about the general concepts involved, consult the Projections Tutorial and subsequent topics.

- The Projections Readings topics provide a classical introduction to projection concepts together with a guide to selecting projections based on the writings of John Parr Snyder, one of the greatest enthusiasts of map projections ever.

- The Manifold Projections topics discuss the specific characteristics in summary form of various families of projections available within Manifold System. This is a reference section that should be consulted for summary information on a specific projection.

- A topic related to projections is Georegistration, the process of matching an image or drawing to the geographic location and projection of a "known good" reference drawing, image, or map. If you wish to use an image within a geographic map please read the Georegistration topic.

- Experts may specify custom units of measure, ellipsoids, datums and even customized coordinate systems (projections) by specifying customized projection presets. See the Customization topic.

- **Very Important:** Do not use the Edit - Assign Projection dialog in an attempt to change the native projection of a component. This dialog is used only to complete the import of a projected component from a format that does not
correctly save projection information. Instead, use the Edit - Change Projection dialog to re-project a component.

- Maps are a special case because they are viewports that show their contents in whatever temporary projection is desired. They don't change their contents, they simply show them in a different form. The projection used by a map to display its contents is set in the Edit - Assign Projection dialog.

**Why Use Projections?**

In Manifold, projections are used for three main reasons:

- To provide a more natural looking map,
- To enable measurements of areas and lengths in printed maps.
- To enable easy measurement in linear units such as meters when performing analyses.

**Better Appearance**

Geographic projections are confusing to many people but the idea is quite simple. Because the Earth is a round, three-dimensional body it's not possible to show the shapes of large entities such as continents on a flat sheet of paper or a flat computer monitor without distorting their shapes. Anyone who has tried to flatten a deflated basketball knows that a rounded 3D surface cannot be perfectly flattened without distorting shapes on the surface.

"Geographic projections" are simply ways of transferring the shapes of things from a sphere onto a flat sheet of paper (or, the same thing, onto a flat computer monitor screen) in a way that minimizes distortions. The idea is to show an image that looks like what one would perceive if one were looking at a globe.

The simplest "projection" is sort of a non-projection: take the longitude and latitude coordinates of objects and plot them as if they were X and Y coordinates. This is a "cylindrical" projection of sorts. Along the Equator and middle latitudes, the distortion is not so bad. More northerly objects are very distorted. Manifold uses this "projection" by default because it is the simplest conceptually. If we are working with small-scale maps such as a map of a particular county near the Equator, we might never need to use another projection. However, if we are working with larger maps such as the entire United States or all of Europe we will probably wish to convert the map to a projection that shows the region in a closer approximation to the way it would appear if seen on a globe.

It is very important to note that no projection can show the entire world without distortion, and every projection to a greater or lesser extent causes some distortion within the map. All projections are therefore approximations. Various projections have been designed for different purposes so when they are used for the intended purpose they do a better job of minimizing distortions that are important for that purpose. For example, some projections are better at minimizing distortions when used to display broad, rectangular regions such as the continental United States. Other projections are better at displaying circular regions such as the Antarctic continent or oceanic coverage of the entire Pacific Ocean.

See the readings in projections provided in this documentation, including General Projections Concepts, The Earth as an Ellipsoid and Guide to Selecting Map Projections for more information on various projections and their uses.
**Accurate Measurement**

Projections are also used to create flat maps from which measurements can be made. Many people are familiar with the idea of taking a ruler to a paper map and measuring the distance between two points, perhaps by comparing the measured distance of a pencil line to a scale printed at the edge of the map. This is a simple and straightforward idea when the map shows a very small rectangle of the Earth’s surface because in such cases even a clumsy projection does not introduce distortions that are greater than the intrinsic accuracy of the map.

When maps cover larger areas of the Earth one must be more careful to choose a projection for the map so that the distortions introduced in the process of projection do not make it impossible to measure distances in the map reliably. The larger the area, the more difficult it is to draw a rectangular, flat map that will enable accurate measurements of length and area to be made at all locations in the map.

The usual solution is to use a projection that is mathematically arranged so that the area of a figure drawn on the globe (such as a country) is the same as that of the equivalent figure on the flat, rectangular map within the area of interest. If lengths are more important than areas, one chooses a projection that does a good job of preserving accurately measurable lengths in the region of interest. If measurement is not required because the map will be used for "thematic" presentation, one is free to use a projection such as the Robinson that allows no accurate measurement at all but that nonetheless delivers a pleasing aesthetic impression.

For work done on the computer screen Manifold will automatically compute areas and lengths of items that appear on screen so one can use whatever projection is desired for visual effect without losing the ability to measure lengths or areas. Because Manifold knows what the true Earth coordinates are supposed to be of every map, the system can re-compute on the fly what the correct areas and lengths should be if the items were laid out on the real, three-dimensional, ellipsoidal Earth.

For measuring lengths using the Tracker tool, Manifold will always compute the tracker length by great circle distance, that is, a straight line between the two points indicated as laid out over the surface of whatever Earth ellipsoid is currently specified as the map datum.

Projections are still important when printing maps because users may attempt measurements from the printed map. In such cases it is important to pick the projection which best preserves lengths or areas in the intended usage.

**Projecting a Map**

Once drawings have been imported into Manifold it is easy to show them in a map and to show the map within any projection desired. To do so, we open the map and use the Edit - Assign Projection dialog to change the projection to any projection desired.

Note that even though a map can show its layers in whatever projection desired, re-projecting the layers on the fly if their components are in a projection different from the one used by the map, at some point the reprojection on the fly process will take so long that users will be unhappy with the resultant performance of the map window for panning and zooming. We can make the map window work faster if both the map window and any large components it contains use the same projection.

See the Projecting a Map topic for more information.
Projections and GIS Formats

Manifold can work with data saved in both projected and unprojected coordinate systems from all standard GIS formats. Unprojected maps are usually easy to import automatically no matter how geographically unaware the GIS format. GIS data saved as projected maps might require user intervention in some cases if the format does not save the projection parameters necessary to use the data.

Once a drawing is imported, the projection parameters in use for that drawing may be seen at any time by viewing its coordinate properties. Click on the drawing in the project view to highlight it and then choose the Edit - Assign Projection dialog for that component. The coordinate properties tell Manifold how to interpret the data in that drawing. If these properties are changed, they simply change how Manifold uses the existing data. Changes in this dialog do not change the data itself. To change the native projection of the drawing, we use the Edit - Change Projection dialog.

Some GIS formats are "smart" and automatically save the projection parameters in use together with the data. During import, Manifold will fetch all necessary parameters from such "smart" formats automatically and will load the coordinates properties with the correct parameters necessary to use the data. Cool!

Other GIS formats (such as ESRI .shp "shapefile" format as commonly used) are legacy formats and do not save the projection parameters in use. When legacy formats are used to save projected maps they require users to keep track manually of the projections used in the file. This is usually done by accompanying the GIS data with a readme.txt file or other "metadata" documentation that specifies what projection parameters should be used with the file. Users must read the file's documentation to know what parameters to use. If we download the file and neglect to fetch the documentation that says how to use it we are in trouble. Ugly!

Since no one wants the hassle of keeping track of projection parameters manually, most people will use legacy formats only to save unprojected data. GIS data in legacy formats therefore is usually unprojected data that may be imported into Manifold with no special user intervention.

On those rare occasions when we encounter a projected map saved using a legacy format there is usually no warning within the format that the data it contains must be interpreted in a way known only to its author. When importing from legacy formats we should always keep in mind the possibility that the file being imported contains projected data. If we suspect projected data is involved we should keep an eye out for a readme.txt or other "metadata" documentation file that might explain what projection parameters are to be used with the file.

To use projected data from a legacy format we first import the drawing using default settings. We then manually add the projection parameters required. To do this, click on the new drawing to highlight it in project view, and then open the Edit - Assign Projection dialog. This will display the (wrong) parameters used during default import. We then enter the correct parameters intended to be used with that data. We only need do this once and then forever more Manifold will keep track of all projection parameters for us automatically.

Note: It's always a good idea to download any accompanying documentation when downloading GIS data from the web even when "smart" formats are used. This will help us speak accurately about the sources of data being used in our projects.

Verifying Projections
Once a component has been correctly imported into Manifold, the system will automatically manage projections from then on. The catch is getting a drawing or image or other component correctly imported into Manifold in the first place if it must be imported from a format that does not reliably capture projection information. Surprisingly many formats for drawings and images used in GIS work are very old formats that do not reliably capture projection information.

It is so important to make sure that a projection has been correctly assigned that Manifold tries to ascertain when a component is imported from a format that does not store projection information. In such cases, Manifold will take special measures to ask us to verify the projection assigned. The first time such a new component is opened, Manifold will overlay an info bar onto the window telling us that the projection used by the component has not yet been verified, and inviting us to click on the info bar to verify the projection.

Clicking on the info bar will launch the Edit - Assign Projection dialog. We should take a moment to review the settings shown by that dialog to verify that the projection is indeed what it should be. If it is, we can click OK and thereafter Manifold will not nag us about verifying the projection. If the projection is not what it should be, we can specify the desired projection. Once we verify the projection assigned, we can open the component without Manifold nagging us to verify the projection.

We can get rid of the info bar without launching the Assign Projection dialog by clicking the X sign at the right side of the info bar; however, Manifold will know that we've not yet verified the projection of this component and the info bar will appear again the next time the component is opened in a window.

If we do not want to verify projections of new components, once we acquire enough expertise not to want to verify projections anymore unless it is absolutely necessary, we can get rid of the info bar by changing the Prompt to verify projections of new components option in the Tools - Options dialog.

Conversely, if we would like Manifold to remind us to verify the projection of all new components, even those imported from formats that correctly store projection information, we can uncheck the Suppress prompt for non-default projections option in the Tools - Options dialog.

**Note:** The rule of thumb used by Manifold to decide if a component has been imported from a format that does not store projection information is to simply raise the info bar whenever a new component's projection is Orthographic, the default used for imports from formats that do not provide coordinate information, or if the component has been imported in Latitude / Longitude with coordinate locations outside the expected range (+/- 90 latitude and +/- 180 longitude).

That leads to some unnecessary use of the info bar as occasionally we will import a component from a perfectly expert GIS format where the component just happens to have truly been stored in Orthographic projection, but for a short hand rule of thumb it works surprisingly well.

More worrisome is that the rule of thumb will miss occasions when formats have some projection information but not complete projection information. The classic case is shapefiles using incomplete .prj accessory files or some image formats accompanied by "world" files. In that case, there may be enough information to cause a projection other than Orthographic to be used, so that the rule of thumb does not cause the info bar to be raised, but not sufficient information from the format to have all projection parameters correctly assigned. In that case, we have be alert enough as users to manually launch the
Assign Projection dialog to verify the projection and to accurately specify all parameters if needed.

Beginning GIS users working with formats such as shapefiles or graphics formats using "world" files may want to uncheck the Suppress prompt for non-default projections option in the Tools - Options dialog so that they are reminded by Manifold to always verify projections on all new components.

Changing Projections of Drawings and Images

Changing the coordinates properties of a drawing or image simply changes how Manifold interprets the internal coordinate numbers that comprise the actual data set. The only time we would do this manually is if we import a projected drawing or image from a legacy format and need to tell Manifold (manually) what parameters are supposed to be used. When importing a projected map from a legacy format, the internal coordinate numbers are already correct but there is nothing in the format that can be imported as correct properties. We must add this information manually.

Changing the native projection within which the coordinate numbers are computed requires us to change the actual coordinate numbers themselves. To re-project a drawing or image into a new coordinate system we open the drawing or image and use the Edit - Change Projection dialog. The dialog opens with the projection parameters currently in use. If the drawing was imported correctly these parameters will be the exactly correct parameters required for the coordinates required in the drawing to make sense.

When we use the Edit - Change Projection dialog to change these properties to cast the drawing into new projection, Manifold will re-compute the coordinates within the drawing to the equivalent numbers required by the new projection. All of this happens automatically. Henceforth, when we open either the projections dialog we will see the new parameters to match the new coordinate numbers. Re-projecting a drawing in this way permanently changes the internal coordinates used to define the drawing.

Because drawings are normally seen through map view within maps, it is not normally necessary to re-project them from whatever projection they were in when imported. Specifying the projection used by a map does not change any data - it simply changes the way the data is seen by computing "on the fly" how the drawing should look in the projection requested of that map view.

The only time we would re-project a drawing is to speed up map view. Map windows works much faster if the drawings they contain use the same projection requested of the map. This is especially true if many, large drawings and images are used in the map. If we re-project a drawing into the same projection used by our map we should make sure that the projection used for the drawing also uses the same parameters as the map. For example, using the same azimuthal projection with the same center latitude and longitude but using different datums is not using the same projection.

Very important: The above discussion of Edit - Change Projection assumes the drawing was correctly imported in the first place. A regrettable trap for the unwary is the use of some older formats, such as ESRI .shp or AutoCAD .dxf, to publish GIS data in projected form. Because these formats do not save projection information, the user must manually enter such information using the Edit - Assign Projection dialog. If you are importing projected data from such formats you must use the Edit - Assign Projection dialog to specify the correct information the format is unable to provide. Until you provide such information the drawing may appear visually correct but it is not yet correctly imported.
Formats such as ESRI .shp or AutoCAD .dxf should not be used to publish projected data but regrettably this is done all the time. See Projections and Legacy Formats for more information in this area.

Images

Although idea of projection is the same in both cases we tend to work differently with images and projections than we do with drawings and projections.

- Images are often re-projected to achieve greater speed in a map view, because images usually contain many more pixels than drawings contain objects.
- Images are usually highly localized views shot from directly overhead or very nearly so. They thus are already in Orthographic projection or are very close to Orthographic. In contrast, most drawings or imported maps are in Latitude / Longitude coordinates and are not in any specific geographic projection.
- Images are often slightly off Orthographic projection, or they are shot at angles that result in a Tilted Perspective or Space Oblique projected view in the image; however, it is very rare that we have actual numeric parameters corresponding to a particular image.

Manifold deals with the above by providing a collection of interactive tools for simple image movement and distortion to transform an image into a correct projection by matching the image to a known good drawing. This process is often called georegistration and allows the use of images with drawings and surfaces in correct geographic position.

A Projections Strategy

For most interactive editing and map preparation work in middle latitudes a good strategy is to keep all drawings in Latitude / Longitude form and to work with them in maps using the Latitude / Longitude projection. This is the name given to the "unprojection" that uses simple degrees latitude and longitude instead of any other projection. Most GIS data is published in this form, so it’s likely that this will be the native “projection” used by most drawings on import.

Use the map in Latitude / Longitude projection for most interactive editing. For fine-tuning the appearance of the map for presentation purposes, create a map that shows the project in whatever projection is desired for presentation.

Maps involving images such as aerial photographs can be conveniently kept in Orthographic projection because the images by default will be in Orthographic projection and because Orthographic provides a typical "view from space" appearance that most people associate with overhead imagery. If speed is an issue, the drawings involved in the map should also be re-projected into the same Orthographic projection.

One limitation of Orthographic is that the projection is defined only for the sphere datum. If other datums are to be used, an alternative "view from space" projection is the Stereographic projection.

If you have an especially fast system or are working with smaller maps you can simply keep drawings in whatever projection they were imported and use whatever projection you like in maps.

XML Files Created upon Export and Used on Import
Not all formats to which Manifold can export have the ability to correctly save projection information. As a safety measure to ensure that projection information is never lost, Manifold always writes an accompanying .xml file when exporting drawings, images, labels, maps or surface components to a file. The accompanying .xml file contains coordinate system information and some other metadata. The XML file is created for all formats, even those that correctly save coordinate system information.

When Manifold imports drawings, images, labels or surfaces, the system will check for an accompanying .xml file that might have been written by Manifold. If such an .xml file exists, Manifold will read it and use the information it contains to load a correct coordinate system.

In addition to the .xml file that is automatically created, Manifold projections dialogs such as the Edit - Change Projection dialog provide toolbar commands to manually write or read projection information from accessory files using generic XML, Golden Software GSR or ESRI PRJ files.

**Automatic use of Custom Datum Transformations**

Manifold uses high accuracy coordinate transformation mathematics when re-projecting data from one coordinate system to another. In certain parts of the world, custom transformation formulae are routinely used (and, at times mandated by law for certain uses) to convert datums during re-projection. If the **Use custom datum transformations** option is checked in the Tools - Options - Miscellaneous pane (checked by default) Manifold will use such custom datum transformations when available.

Currently, NADCON formulae are used to convert between NAD27 and NAD83 in North America and NTv2 formulae are used to convert Canadian, Australian and New Zealand datums supported by NTv2.

If a converted location is outside of the conversion domain supported by such custom methods, no datum conversion is done and Manifold will display a message box to that effect after the attempted re-projection. If desired, such a re-projection can be undone, the data set edited to fall entirely within the region supported by the custom method and re-projection attempted once more.

**Note:** the message box mentioned above appears only when a component is re-projected using **Edit-Projection**. It does not appear when a component is re-projected on-the-fly for display as part of a map view.

In general, if we attempt to make a datum conversion between any of the datum pairs shown below and the datum is not converted, that tells us part of the data set we are attempting to convert falls outside the service area of the custom transformation formulae.

**Datum Pairs supported by Custom Transformations**

- North American 1927 (mean for CONUS) &gt; **NADCON**
- North American 1983 (mean for CONUS)
- North American 1927 (mean for Canada) &gt; **NTv2**
- North American 1983 (mean for CONUS)
- Australian Geodetic 1966 (Australia, Tasmania) &gt; **NTv2**
- Australian Geocentric 1994 (GDA94)
Custom datum transformations will be invoked only when converting between the above pairs in either direction. For example, they will be used when converting NAD27 to NAD83, or when converting from NAD83 to NAD27. Pairs shown above are listed using their full names as they appear in the datum box in Manifold projection dialogs, not using the abbreviated form commonly used. For example, North American 1983 (mean for CONUS) is referred to as NAD83 in abbreviated form.

The names of the above datums are significant, not the parameters used. For example, if we convert GRS80 (the parameters of which are identical to NAD83 so it is the same datum) to NAD27 (mean for Canada), Manifold will not use the custom NADCON conversion formulae. This is done on purpose, so we can use NAD83 as the name of the datum for conversion-sensitive North American data (so that the custom transformation will be used) and GRS80 as the name of the datum for all other data where we would like to use the general purpose routines.

Custom Datum Pairs for NTv2

Additional datum pairs may be added to those supported by Manifold for use in NTv2 transformations through customization. See the Custom Datum Grids for NTv2 topic for details.

A Technical Note on Datums

This technical note is intended for experts.

The most commonly used datums for many purposes (such as the default datums used by GPS devices) are WGS84 worldwide and NAD83 in North America. While in almost all cases these datums may be thought of as fixed, well-defined datums not subject to change, that is not true in all cases as sometimes the definitions of those datums will vary from the "standard" definitions embedded within Manifold's XML file definition and as used by virtually all GIS applications and almost all hardware (such as most GPS devices).

In actual fact both the WGS84 datum and the NAD83 datum change slightly over time to account for changes such as those caused by movements of tectonic plates. The working definitions of these datums as used in some high-precision applications will therefore change from the "standard" definitions of these datums.

To use a working definition for either WGS84 or NAD83, create a custom datum from the working definition and use that instead of the WGS84 or NAD83 datum built into Manifold. Such working definitions are not published within this documentation or with Manifold because the point of those definitions is that they change over time. Therefore, the user must find out exactly how such "working" datums are defined within the particular usage that employs them. This can be a major annoyance, especially if someone uses such an atypical definition for a very common standard like WGS84 without alerting users that something unusual is going on.

The need for such custom datums usually arises when working with GPS devices that employ working definitions of such datums instead of the prevailing standard definitions.

For example, certain Trimble devices equipped with WAAS will use a working definition of NAD83. To use data from such devices in Manifold, first find out what working definition is used by that device. Create a custom datum from that working definition using a distinct name of your choosing, such as "NAD83T."
Next, create a drawing in latitude / longitude using the standard NAD83 datum. Accept data from the Trimble device into that drawing. To correct the coordinate system used, launch the Edit - Assign Projection dialog and change the datum to use your custom NAD83T datum.

Note that some Trimble devices not equipped with WAAS will use a working definition of WGS84 instead of the "standard" WGS84 definition. These may be dealt with by using a custom datum created from the working WGS84 definition used by Trimble.
Projecting a Map

This topic assumes all we want to do is project a map using drawings or images that have been imported into Manifold correctly. When using drawings or maps in Manifold format or other modern GIS format that correctly saves information about projections there is no need to understand the internal workings of projections. There are also few difficulties when we import unprojected data.

If you are struggling with the import of a projected map (say, in UTM or State Plane Coordinate System) using data in some ancient format such as ESRI .shp, please jump to the Projections and Legacy Formats topic and then follow the links to all the ugly stuff you must consider.

To project a map, we take note of some map characteristics and then we use the Edit - Assign Projection dialog.

Step 1: Note the Center of the Map

Most projections require a center for the projection. That's the latitude / longitude location at the center of the region to be projected. Before launching the Edit - Assign Projection dialog we should note the center and some other parameters in case we want to set projection parameters manually. Manifold has a Suggest button in the projection dialog that will set parameters for us but it is often helpful to note them in advance so we can compare the suggested parameters to what we might choose to set on our own.

To determine the center of the map move the mouse cursor to the center of the region and note the latitude / longitude numbers given in the status bar. For larger regions, round up to some reasonably even number. For example, a reasonable center for maps of the US made in Lambert Conformal Conic is 40 degrees latitude and -100 degrees longitude.

Step 2: Choose the Right Projection

Launch the Edit - Assign Projection dialog and choose the projection desired. The projection to use depends mainly on the region to be mapped and the intended purpose of the map. The suggestions below work well in almost all cases and are based on the region to be mapped.

There are so many projections available within Manifold that they are organized within a tree diagram. All of the following are in the Standard category. The subcategories are noted in parentheses where the name of the projection itself does not make it clear to which subcategory it belongs.

Most users will use only a few projections over and over. For general use, the Manifold team recommends the following:
Region to be Mapped | Suggested Projections
--- | ---
Whole Earth | Robinson (pseudocylindrical) or Miller Cylindrical. Robinson seems to be fashionable for thematic maps. Any of the pseudocylindrical projections will be fine if you like their appearance better.
Hemispheres | Orthographic (azimuthal) for a "view from space" look, and Lambert Azimuthal Equal Area for thematic maps where the relative size of countries near the edge of the projection is to be preserved.
Continents | Use Lambert Conformal Conic for North America and Eurasia. Use Lambert Azimuthal Equal Area or Orthographic for South America and Africa. Use Orthographic for Australia, and Antarctica.
E-W Countries or Regions | Use Lambert Conformal Conic for US, Canada, Russia, and China. Use either Lambert Conformal Conic or Orthographic for Europe. Use Orthographic or Lambert Azimuthal Equal Area otherwise.
Polar Regions | Orthographic or Lambert Azimuthal Equal Area.
Oceans | Orthographic or Lambert Azimuthal Equal Area.
Smaller Countries or Regions | Orthographic.
N-S Countries, Oblique Regions | Long, thin countries aligned North-South such as Chile are one of the few times we would use Transverse Mercator. Oblique regions like the Alaska panhandle are mercifully rare: Use the Oblique Mercator in such cases.

Other than personal taste in visual appearance there are only three reasons not to use one of the above projections:

- A specific projection is required to match other data available or to exchange data with other users.
- The technical characteristics of a particular projection such as preservation of scale or relative area are required for creating a paper map or other presentation.
- A specific technical characteristic, such as datum used, is not a a good match to the data.

As an example of the latter point, one limitation of Orthographic is that the projection is defined only for the sphere datum. If other datums are to be used, an alternative "view from space" projection is the Stereographic projection.

Manifold provides a very wide array of projections in addition to the standard projections mentioned above.

**Step 3: Project the Map**
In the **Edit - Assign Projection** dialog specify the projection selected using the tree diagram and the center for the projection and other parameters. When specifying latitudes and longitudes remember that **west** longitudes are **negative** numbers and that **south** latitudes are also **negative** numbers.

Some projections will require other parameters. For example, when using Conic projections two "standard" latitudes (also known as "standard parallels") are required. These should be about one fifth of the way down and one fifth of the way up the map for greatest visual fidelity, but most people will round these values to memorable numbers. For the US, many people use 35 degrees as the first standard latitude and 45 degrees as the second standard latitude. The **Suggest** button will automatically load reasonable values for standard parallels into the projection's parameters.

The **Suggest** button will cause the system to estimate the center of the map based on the size of the largest drawing rectangle that participates in the map. Manifold will then load the estimated, rounded center into the **Center Latitude** and **Center Longitude** parameters together with estimates of "reasonable" values for the other parameters. These may then be edited as desired.

There's no need to worry about the **Datum** unless we are exchanging projected data from old formats or are using Manifold to convert GIS data between different projections or datums. If we use **World Geodetic 1984 (WGS84)** as our standard datum we'll be in great shape to utilize GPS data and other modern instrumentation, almost all of which use WGS84.

If we need to match a particular projection used by other data or required for data exchange we must find out what datum is necessary and pick it from the list of datums available. Manifold includes virtually every datum that has been used anywhere in the world for mapping.

The projection dialog includes several options for use by experts. These are changed only in special cases, such as manual georegistration. See the **Edit - Change Projection** dialog topic.

**Additional Notes on Choosing a Projection**

For showing small regions of the Earth, such as a hiking trail within a park almost any projection is acceptable. Any distortion of shape, area or scale for such small regions caused by the projection will likely be less than the intrinsic accuracy of the data. In small regions almost all applicable, correctly centered projections are visually indistinguishable even to experts.

We suggest the Orthographic projection because it provides a natural appearance of mapped shapes from an overhead view. It is also the "default" projection used with images and abstract CAD drawings when they are first imported into Manifold. This will enable fast usage of images within maps when all use the same projection.

For mapping larger regions, such as countries or continents, learning about a handful of frequently used projections is enough. The main choice is between projections that provide better appearance and accuracy of East-West regions (like the US), North-South regions (like South America) or circular regions (like Antarctica or the Pacific Ocean). In almost all cases a choice between Lambert Conformal Conic (for East-West regions) or Orthographic (for South America or Africa) is a good choice. The Lambert Conformal Conic projection uses two additional standard latitudes as described in the Lambert Conformal Conic projection's topic.
For showing the entire world at once, there is no single projection that adequately minimizes distortion while preserving accurate representations of area or scale. Maps of the entire Earth are used almost exclusively for thematic presentation purposes and not for measuring purposes such as navigation. In such maps the main objective is a visual appearance that preserves in a subjective way the relative sizes of countries, with rough preservation of shapes.

Most users will gravitate toward one of the standard cylindrical or pseudocylindrical whole-world projections and will tend to use it over and over. It is almost exclusively a matter of personal taste. We suggest either the Robinson because it is currently a fashionable projection or the Miller Cylindrical because it provides straight lines for parallels and meridians. Many users will prefer the Mollweide or another choice.

The only projection for whole Earth use one should absolutely, positively avoid is the Mercator projection. The Mercator projection was intended for marine navigation. Its main positive feature is that it has straight rhumb lines, an essential requirement for marine navigation. As a whole Earth projection it is one of the worst possible choices due to the great distortion of regions not near the Equator.

**Why So Many Projections?**

If only a handful of projections are recommended, why does Manifold provide so many other projections? Manifold provides so many projections because many projections are in common use. To import data from those many different projections Manifold must understand them all.

Many projections are in common use because most mapping of the Earth was done before the era of fast and cheap computers. In earlier times paper maps were the living instruments used to measure the Earth. The specific technical characteristics of projected maps such as scale preservation are extremely important if paper maps using such projections will be used as measuring instruments. Different map projections were required for different countries and regions so that mechanical measuring and plotting on paper maps for different purposes could be easily achieved.

With modern computer software none of that is necessary any more. Measurements of distance or area within a GIS on a computer can be done with far greater precision than by using a straight-edge and compass on a paper map. As a result paper maps are less frequently used as precision measuring instruments. In modern times paper maps are used for presentation and for visual estimates of relationships.

There is no longer any reason to restrict our projection methods because they are easy to compute by hand. For presentation purposes computers can instantaneously create a custom-centered projection of a data set to suite whatever specific view is required. With computers and software like Manifold we can use relatively few different map projections and simply set the parameters so the projection is always a good match for the view desired. However, since the world's database of digital maps was created mainly from the legacy archive of paper maps the numerous different projections created for use with paper maps continue to be prevalent in the digital era.

This is not necessarily a negative thing, since different projections provide different artistic impression and even a different conceptual feel. Having a rich set of projections from which to choose is like having a nice collection of fonts to use in one's text documents. Different projections can be used to good effect for improving the visual appearance of presentations.
For more on choosing projections, see the Projections Tutorial, General Projections Concepts and the Guide to Selecting Map Projections.

For more on datums see The Earth as an Ellipsoid.

**Re-projecting Drawings, Images and other Components**

The above procedure describes re-projecting a map. To permanently re-project a specific component we open it in a window and apply the Edit - Change Projection command. If the component appears in a map and we would like to re-project it to use the same projection as the map, we can right click its tab in the map and choose **Project to Map**.

**Differences between Assign Projection and Change Projection**

- **Assign Projection** appears both with map windows and with individual component windows. When used with map windows it changes the projection used in the map view without actually changing the data in the components. Map windows have dynamic projections that can show their contents in whatever projection is desired regardless of the native projections of what is inside them. When used with component windows, this dialog changes the projection assigned to be used to interpret the coordinate data within the component.
- **Change Projection** appears only with individual component windows such as drawing windows or image windows. This command changes the actual coordinate data in the component into a new coordinate system. It can only work correctly if the starting projection assigned to that component was correctly assigned, so before we can use **Change Projection** to change a projection we must use **Assign Projection** to verify that the projection originally assigned was correctly assigned.

**Problems with Drawings from Legacy Formats**

Failing to import a projected drawing correctly will lead to unpredictable effects when the drawing is used in a map. If we import a projected drawing from a geographically unaware format such as `.shp` or `.dxf` we must tell Manifold the correct projection to use to interpret that data since the format does not save this important information. It is, of course, a blunder to publish projected drawings using formats such as `.shp`, but this is done all the time.

To import a projected drawing or image from a geographically unaware format:

1. Import the drawing or image using **File - Import**. Use default settings.
2. Open the drawing or image.
3. Use the Edit - Assign Projection dialog to specify the projection information that should be used.

The **Edit - Assign Projection** dialog allows us to manually provide the correct projection information into the coordinates properties. Manifold will then be able to make sense of the data just imported.

Do not confuse the use of a chosen projection view in a map window (via Edit - Assign Projection) with the use of Edit - Assign Projection in a drawing window to specify projection information missing from legacy formats. If a projected drawing is imported from, say, `.shp` format and you fail to tell Manifold the correct projection to use, that drawing has not yet been correctly imported. Subsequent use of the drawing in maps may
cause bizarre effects or lengthy delays as the system attempts to compute a new projection for the map based on fundamentally inaccurate coordinates.
Projections Quick Reference

This topic provides a summary of key coordinate information for experienced GIS users.

If you are a new user and are working within Manifold System only, jump to the Projections topic and the Projecting a Map topic. Very little expertise in the inner workings of projections is required to make simple projections when working within Manifold System.

If you must deal with maps in legacy formats such as ESRI .shp or AutoCAD .dxf, please read the Projections and Legacy Formats topic. Because legacy formats do not normally save information on projection parameters used, importing maps saved in legacy formats can require extensive knowledge of projections and coordinate systems.

If you are new to projections and coordinate systems in GIS or would like a refresher on coordinate systems and projections consult the Projections Tutorial and subsequent topics followed by a quick return to this topic as a summary.

- Manifold maintains all coordinates using double-precision, floating-point numbers. This allows mapping the Earth with accuracy to 16 decimal digits of accuracy. If we divide a meter (about a yard) into two hundred and fifty million parts, Manifold's accuracy is better.

- A map shows any combination of drawings or images reprojected "on the fly" into whatever projection is used in that map. Use Edit - Assign Projection to specify the projection a map uses. This changes the projection used by the map without changing the native projections of any drawing or image used in the map. Because a map window allows multiple layers it is the preferred window for work in Manifold.

- Drawings and images are shown in drawing windows and image windows using their native coordinate systems. Drawings and images may be reprojected so their internal coordinates are already transformed into a specified projection. If a particular projection will be frequently used with drawings or images in a map it is wise to convert them into that projection so there is no need to reproject them on the fly.

- Use Edit - Change Projection to transform the coordinate numbers inside a drawing or image into their equivalents for the desired new projection. This is a permanent alteration in the internal coordinates used to create a drawing or image. This dialog allows export of projection information to XML, Golden Software GSR or ESRI PRJ formats (as well as import from those formats) to allow for easy creation of accessory files when exporting components for use in other software package. The dialog also allows quick fetching and reuse of coordinate information specified for other components, to avoid having to manually re-enter coordinate information whenever possible.

- Images that originate in raster data are normally not re-projected since doing so invariably adds or removes pixels via interpolation. Instead, raster data images are re-projected on the fly in Map view to achieve the desired effect. In contrast, images used purely for visual effects are often re-projected using the Edit - Change Projection dialog so maps that use them will display rapidly.

- CAD-style drawings created using linear coordinates such as meters or inches are interpreted by default as using the Orthographic projection centered on the 0,0 origin of the Prime Meridian and the Equator. If not otherwise georeferenced, images also are placed with their lower left-hand corner at that geographic 0,0 origin and are taken to be in Orthographic projection. Manifold's internal accuracy is so high that
images scanned at over 4800 DPI can have a distinct geographic location for each pixel.

- Since images shot from aircraft or satellites are often overhead views of very small portions of the Earth, they may be conveniently used within the default Orthographic projection used for images. This allows them to be used in a casual way in map layers with cartographic drawings that also use Orthographic projection centered on the current view. This is a fast setup where Map view can operate without any re-projection on the fly. We can use the various image georegistration tools to adjust such images so they are a good match to the vector drawings.

- All drawings using projections other than degree-based "Latitude / Longitude" are based on linear measures, usually meters, centimeters or feet as units of measure. When importing from a "smart" format the coordinates properties for such drawings will be automatically set by Manifold during import.

- Legacy formats like ESRI "shape files" are not normally used for saving maps in projected form. If we have the misfortune of encountering a projected map saved in a legacy format we must make all efforts to locate and grab any documentation that accompanies the file to see what projection parameters were used. We will need this information to tell Manifold what to do with the numbers being imported. After such maps are imported into Manifold as drawings, we must manually change the coordinates properties for that drawing using the Edit - Assign Projection dialog to specify the correct projection parameters based on what we read in the metadata. Note that even if the shapefile is in unprojected Latitude / Longitude coordinates we will have to specify which datum is used using this dialog.

- The Edit - Assign Projection dialog is normally used only once when importing projected drawings from legacy GIS formats. Requisite parameters may be manually added with this dialog after importing the drawing. Changing the projection properties of a drawing does not change the raw coordinate numbers that define the drawing's objects. It simply changes the way the numbers are interpreted by the system.

- It's often the case that images used in mapping will not be a perfect overhead view of the subject. We will often receive such images in geographically-mute image formats such as .jpeg that have nothing to say about the projection used. Technically, such images are often in Tilted Perspective or Space Oblique projection but with unknown parameters. In Manifold, we deal with such images by assuming they are Orthographic and then we use Manifold georegistration tools to warp the image to fit a "known good" vector drawing or set of lat/lon coordinates. In effect, we are re-projecting the image into Orthographic by nudging its scale and position while warping it. This georegistration process is usually easy and very satisfactory in the case of near-overhead images.

**Very Important**

To re-project a component, use the Edit - Change Projection dialog. Do NOT use the Edit - Assign Projection dialog. This latter dialog is a specialized dialog that is used to manually correct the projection assigned to a component imported from a format that is unable to provide projection information.

**Customization**
Experts may specify custom units of measure, ellipsoids, datums and even customized coordinate systems (projections) by specifying customized projection presets. See the Customization topic.
**Projections and Legacy Formats**

Projections are used in maps for two main purposes:

- To provide a more natural looking map, and
- To enable measurements of areas and lengths in printed maps.

Projections can be extremely simple to use, with little knowledge required if one works exclusively within Manifold using digital maps saved in modern formats. Using Manifold, one can make pretty projections with a point and a click. It's fast and it's fun.

To project a map, use the Edit - Assign Projection command to choose whatever projection you like. The map will show its contents in that projection no matter what the "native" projections of the components it contains. To change the projection used in a drawing, image or surface permanently, open the component in its own window and use the Edit - Change Projection dialog to re-project it. If that's all you would like to do, read the Projections topic and then jump straight to Projecting a Map.

The situation becomes considerably more complex when importing data into Manifold that was created using older GIS systems and saved in formats that are not well suited for publishing maps. Complexity also rears its ugly head when importing geographic images and surfaces from formats that do not save projection data. We can use such data in Manifold, but to do so we will have to learn more about projections and follow a more complex import process.

Most digital map data is published in unprojected form to assure the greatest possibility of accurate interchange between various GIS systems and to reduce hassles for users. Users can then cast the map into whatever projection is desired using the facilities of their GIS package. Occasionally one encounters digital map data published in projected form. Dealing with projected maps that are published in old formats can be a real nightmare.

If you must deal with maps in legacy formats such as ESRI .shp or AutoCAD .dxf you probably will have to learn a lot about projections and coordinate systems. At a minimum, you must learn enough to be able to tell whether the digital map being imported was saved as a projected or as an unprojected map. Because legacy formats are geographically mute when it comes to projection information, users must acquire a considerable level of understanding just to be able to tell if they are dealing with a projected or unprojected map.

The situation becomes even more complex if a legacy format is used to save maps using old-fashioned projections such as UTM. In such cases one can be pulled into a confusing sequence of tasks requiring deep technical knowledge of the inner workings of map projections cast within ancient GIS formats. Not only is it a conceptual mess, it requires learning a quasi-technical lexicon of obsolete words, such as "false easting" and "false northing." It is as if to purchase a hamburger at McDonald's one needed to learn the farming methods and jargon of 1940's cattlemen.

Learning about projections in detail requires learning about coordinates and numerous other technical topics. This is not easy for non-mathematical people. It is especially difficult if the overall concepts are muddied by the tactical hassles involved in importing data created in older software using older formats. For this reason, it is a good idea to begin by learning to use projections exclusively with maps from the Manifold downloads site (start at the Projecting a Map topic and continue with the Projections Tutorial topic).

Once you understand how to use projections within Manifold without the complications of dealing with legacy formats or old-fashioned methods you can dive into extended reading on projections and coordinates topics, beginning with the Coordinates topic.
Older File Formats and Projections

Manifold saves maps in a project file together with all technical parameters required to manipulate the data regardless of what projection is used. One may combine data from such maps in a completely free-form way and Manifold will automatically make all adjustments necessary. In a modern system such as Manifold, everything will always appear in the expected latitude / longitude coordinates everyone understands are used to identify locations on the Earth’s surface.

Older GIS software was not always so flexible, and formats used to create maps in older systems are often unable to store important information about the projection in their file formats. Part of the user-unfriendly nature of older software and formats is that the user is expected to manually keep track of important information such as projection parameters. Quite literally, the user is expected to make notes on paper or in some sort of readme.txt file that accompanies the digital map files so that the GIS can be told what projection parameters to use with that file.

In modern times we recognize that requiring users to manually keep track of which projection parameters were used to create a particular file is a formula for endless hassles and errors. However, older GIS formats that require this of users continue in service within organizations that produce many digital maps we would like to use. In at least one case (ESRI .shp format) a format unsuited for publication of projected maps continues in use as the main format behind the most widely distributed GIS software within US government and universities. In another case, a CAD format, AutoCAD .dxf, is widely used to publish both projected and unprojected maps even though .dxf is highly unsuited for geographic map publication of any kind.

When governments and universities publish maps in unprojected form using ESRI .shp or other old formats there is usually little harm done except for the ordinary limitations of old formats such as limited file names. Even the difficulties presented by AutoCAD .dxf usually involve no more than minor hassles when unprojected maps are involved. In contrast, major problems arise when formats unsuited to the saving of projected maps are used to publish maps in projected form.

When formats like .shp and .dxf are used to save maps in projected form, one must know technical details about the projections used because one must manually provide information the format is unable to save. This introduces another conceptual minefield for users, the need to know enough about the technical internal details of various projections to be able to recognize and write down the necessary key details. This is not always easy, since some projections are themselves relics from the dawn of time. The Stereographic projection, for example, was known to the ancient Egyptians.

Many maps have been created using projections that are reflections of the limitations of the technology of their day and not because they are the best projections to use for a given purpose or are the most user-friendly. The notorious Universal Transverse Mercator (UTM) system, for example, was created in part in recognition of the limits of computation when doing arithmetic with hand-cranked calculating machines. Even though UTM was created before desktop computers were imagined it persists into the present day to wreak havoc among GIS novices. Because many maps were created in such systems and because technologically static governments have adopted some obsolete systems (like UTM) as the basis for legal norms they persist as living fossils into modern times.

If we use digital maps from many different sources or if we wish to exchange mapping information with colleagues using legacy systems, sooner or later we will need to understand projections and the coordinates within them at a seriously detailed technical level. This will be very frustrating for the new user. At times, as with UTM, the bizarre
technical limitations of a living fossil projection presented within a living fossil data format such as .dxf will try the technical patience of even professional cartographers. However, if we persist and gain the technical understanding required we will be able to decode ancient formats we find on contemporary government and university web sites. We will be able to get many free maps and we will then be able to combine map data from such maps with other maps. We will be able to communicate with organizations still stuck in the technological amber of an earlier day.

It must be emphasized that there is no reason why in modern times most users of a GIS system should need to know the internals of how projections and coordinates work. This is all a result of legacy data and legacy formats. Manifold provides a very rich set of tools for dealing with such relics when they are encountered, or for the expert usage of cartographers who desire direct control over all aspects of their data. However, knowledge of internals is only required when working with legacy map data and not because of anything intrinsic to Manifold.

Problems with Drawings from Legacy Formats

Failing to import a projected drawing correctly will lead to unpredictable effects when the drawing is used in a map or other usage within Manifold. If we import a projected drawing from a geographically unaware format such as .shp or .dxf we must tell Manifold the correct projection to use to interpret that data since the format does not save this important information.

Manifold will remind us of this need by always raising an info bar the first time a new component that was imported from a geographically unaware format is opened. The info bar reminds us to verify the projection assigned to that component.

Note: The rule of thumb used by Manifold to decide if a component has been imported from a format that does not store projection information is to simply raise the info bar whenever a new component's projection is Orthographic, the default used for imports from formats that do not provide coordinate information, or if the component has been imported in Latitude / Longitude with coordinate locations outside the expected range (+/- 90 latitude and +/- 180 longitude).

To import a projected drawing from a geographically unaware format:

1. Import the drawing with File - Import. Use default settings.
2. Open the drawing.
3. Use the Edit - Assign Projection dialog to specify the projection information that should be used.

The Edit - Assign Projection dialog allows us to manually provide the correct projection information into the coordinates properties. Manifold will then be able to make sense of the data just imported. The Edit - Assign Projection dialog is normally used only once immediately after importing the drawing to specify correct projection information that the drawing's original format was too stupid to provide. It is not used to re-project the drawing. To re-project the drawing, use the Edit - Change Projection dialog.

Do not confuse the use of a chosen projection view in a map window (via Edit - Assign Projection) with the use of Edit - Assign Projection in a drawing window to specify projection information missing from legacy formats. If a projected drawing is imported from, say, .shp format and you fail to tell Manifold the correct projection to use, that drawing has not yet been correctly imported. Subsequent use of the drawing in maps will cause bizarre effects, lengthy delays or other incorrect operation as the system attempts to deal with the drawing based on fundamentally inaccurate coordinates.
Also, do not confuse the one-time use of **Edit - Assign Projection** with the regular use of **Edit - Change Projection**. Changing the internal understanding of a drawing's coordinate numbers by using **Edit - Assign Projection** will not re-project the drawing. To re-project the component, use **Edit - Change Projection**.

**Notes**

The classic "ancient" formats are ESRI **.shp** and AutoCAD **.dxf**. Neither saves projection information as commonly used. Use these formats for unprojected data only. There is a recent hack to the **.shp** format that will save projection information; however, it is so recent it does not help with the great mass of legacy data already published in **.shp** format. If you have a choice when getting projected legacy data, use **MapInfo .mid/.mif** format. Mid/mif is a reasonably modern format that saves projection information within the map files. Avoid **.dxf** at all costs.

When importing from **.shp** format, even when importing an unprojected shapefile keep on the lookout for any information that tells you what datum was used in that file. Importing an unprojected shapefile will by default import the shapefile using **Latitude / Longitude** projection using the **WGS 84** centroid. Such imports will be acceptable for many GIS purposes; however, for full accuracy one should find out what centroid / ellipsoid / datum (three words with slightly different meanings but often used as synonyms) was used for the shapefile being imported and use the Edit - Assign Projection dialog to specify it for the imported drawing.

The classic "living fossil" projections that confuse new users are UTM, State Plane Coordinate System and Gauss Kruger. If you have a choice between receiving projected data in these systems or receiving unprojected data, always choose the unprojected data.

If you have no choice but to receive data in old formats remember to stay alert for any "readme.txt" file or any other documentation that accompanies the data files. With luck, the author of the data will have provided some notes on projection parameters or other information used within the map.

If you are not sure whether or not a **.shp** format or other old format file is a projected or unprojected map, an easy way to find out is to import it using default settings for the importer. One can then overlay the imported drawing as a layer in a map with a layer imported from a "known good" drawing from the Manifold downloads site. If the imported drawing appears where it should geographically, it almost certainly imported OK and was quite likely an unprojected map. If it does not appear in the correct geographic context then the original file most likely contained a projected map. In such cases, the projection parameters to be used with the file must be tracked down.

Using **.shp** for projected maps and **.dxf** for anything causes such hassles during interchange that one is tempted to say bad things about publishers who do such things. However, it is often the case that the publishers themselves are trapped by organizational politics into using inappropriate formats. In some cases they might not have enough time to convert data from old formats and are simply trying to get data out to the public for better or for worse. If you find yourself in this position as a publisher please do your best to publish **unprojected** data sets. If you must publish projected data sets, please try to provide clear documentation for the projections that were used.

In this topic we refer to "drawings" because most projected data sets saved in legacy formats are drawings. Images and surfaces are also occasionally found as projected data as well. For that matter, images are often encountered that are "whole Earth" images in latitude / longitude projection and published using non-geographic formats such as **.jpg**.
See the Cookie Cutter a Large Image with Transfer Selection topic for an example of how one uses **Edit - Assign Projection** to complete the import of such images.

Some images are accompanied by "world files" that are said (by the uninformed) to contain full projection information allowing automatic import of georegistered images. Regrettably, that is not true: world files contain only partial projection information and still require manual intervention. See the Importing Images topic for a complete discussion.

The use of .prj files together with shapefiles (as a sort of machine-readable "readme" file) to specify projections is a recent development that, unfortunately, has come into play without any definitive standards being published to precisely specify how the .prj should code its contents. Although the use of .prj files works most of the time, it is something of a haphazard system. If a shapefile is accompanied by a .prj file Manifold will analyze the .prj to try to determine what projection and projection parameters are intended for the shapefile.

Manifold recognizes over 500 projection types used in .prj files and so will be able to extract the projection information in almost all cases, reading more different types of .prj than almost all other software packages that can read shapefiles. However, because .prj files do not use standardized nomenclature for projections it is possible that in some cases it will not be possible to read the desired projection even if the shapefile is accompanied by a .prj file. manifold.net is always on the lookout for .prj files containing previously-unknown standards to add to our "zoo" of strange examples. If you encounter a shapefile using a .prj that does not import into Manifold correctly, please contact tech support for instructions on how to FTP it to manifold.net for examination.

**See Also**

Import Drawing - SHP, Shapefiles for an example of importing a projected shapefile and subsequently setting coordinate properties manually.

**Import a Projected Shapefile** for a detailed example.

Import Drawing - DXF for short, but gruesome examples of importing an unprojected and a projected DXF file.
Projections Tutorial
This is the introductory topic on geographic projections for users new to GIS and mapping. It is a good refresher for experienced users wishing to understand how Manifold deals with projections.

Geographic projections are a way of showing the curved surface of the Earth on a flat surface like a piece of paper or a computer monitor.

For over two thousand years educated people have known that the Earth is round and have realized as a matter of elementary geometry that any flat map showing the surface of our curved Earth will in some way change the shape of what it portrays. For over two thousand years geographers have been inventing ways of using flat maps to show the curved surface of the Earth in ways that minimize such distortions.

Imagine we make a globe out of a flexible material with a world map painted onto the material. Now, let’s cut the material and try to flatten it. If you’ve ever tried to flatten a deflated ball you know this is not possible to do without stretching the flexible material in some areas and compressing it in others. If we deform the peeled "skin" of our atlas globe in this way to make it flat we will end up changing the shape of continents and other items illustrated in our world map.

There is no one way of projecting the curved surface of the Earth onto a flat sheet that does not cause some distortion, and there is no one projection that is suitable for all purposes for which people use maps. However, for virtually every usage there are projections that minimize distortions of importance for that task. For example, if one needs to measure areas in a flat map there are projections that will guarantee the area contained by various shapes is correct even if the shapes of the objects shown appear quite differently than they do on a globe. Other projections do a good job of showing continents in a shape similar to that seen on a globe even though they do not allow for accurate measurement of areas.

Note: We refer to the Earth as a sphere in this topic even though it is a slightly flattened ellipsoid.

How Projections Work

Any shape that is curved in only one direction can be unrolled into a flat map without distorting the appearance of objects drawn on it. For example if we take a cylinder and cut it lengthwise we can unroll the cylinder into a flat map. The trick to any elementary projection is to place the Earth within such a shape and to "project" lines out from the Earth onto the shape to show where to draw the projected outlines of items on the globe. We can then "unroll" the shape and see our projected map on a flat surface.
The first and most obvious such projection to use is a simple cylindrical projection. Place the Earth inside a big cylinder that touches the Equator and then transfer points on the globe to the cylinder. The simplest way to do this is to imagine the cylinder is graph paper with 360 boxes in circumference and 180 boxes up and down.

If we use the longitude and latitude coordinates in degrees of a place on the sphere and transfer it to our cylindrical roll of graph paper we end up with a map like the above. Since one point needs to be the center of the unrolled cylinder, nearly universal usage is to use the intersection of the Equator with the zero Meridian running through Greenwich, England. We can then count degrees plus and minus 180 degrees in longitude and plus and minus 90 degrees in latitude.

The above presentation is called the **Geographic** or Latitude / Longitude projection. We can think of it as our default projection. It produces a good effect in areas near the Equator, but results in immense distortion close to the poles.

There are other ways of transferring points from the surface of the globe onto an enclosing cylinder. Most of these, such as the Mercator projection, use some mathematical formula to alter the ratio between degrees of latitude on the globe and vertical measurements on the cylinder. What they all have in common is that accuracy is good near the Equator where the cylinder is very close to the globe. To a greater or lesser degree all cylindrical projections centered on the Equator fall off in accuracy as distance from the Equator increases.

If we wish to make maps of places along the Equator we could use a cylindrical projection and just show those regions. What would be shown in those maps would be relatively free of distortion. One problem with this is that the Equator for the most part lies over water whereas the greatest demand for maps is in populated zones. A quick glance at a world map shows that most populated zones occur in a North-South direction.

Turning the cylinder so that it is tangent to the Earth along a meridian (longitude line) instead of tangent to the Equator results in what is called a **transverse** cylindrical projection. We can now make local maps anywhere along the darker, North-South line of
tangency and if the maps are not too big they will be relatively free of distortion. However, this only works along the line of tangency. If we pick a North-South line running through Athens we can make maps all the way from Scandinavia down the length of Africa, but any maps using this projection in North and South America would be hopelessly distorted.

One possible solution is to use not one projection, but many transverse cylindrical projections with the cylinder rotated slightly along the Equator. In fact, one scheme of mapping the Earth called the Universal Transverse Mercator (UTM) plan does just this. UTM maps the Earth with a transverse cylinder projection using 60 different lines, each of which is a standard "UTM Zone". By rotating the cylinder in 60 steps (six degrees per step) UTM assures that all spots on the Earth will be within 3 degrees of the center, tangent line of one of the 60 cylindrical projections. (The Gauss Kruger system is a European system akin to UTM that also uses a transverse cylinder rotated in six degree steps).

To map any spot on Earth in UTM, one picks the UTM Zone centerline that is closest to it and then makes a map using that cylindrical projection.

The illustration above shows a small section of the earth near the tangent line projected onto the cylinder, and then the cylinder being unrolled into a flat sheet. If we want to save the X,Y locations of points on our flat sheet we can now measure them as though the flat sheet were graph paper and use the resulting coordinates in a digital, flat map.

The above illustration shows a key concept that often proves confusing to GIS newcomers: although "unprojected" data about locations on the Earth are specified in degrees, all projected maps specify the coordinates of the objects on them using X,Y coordinates using meters, feet or other linear measures. These coordinates are computed relative to some origin on the flat sheet established by the projection in use.

Computer files that contain projected maps therefore contain coordinates like
and not longitude, latitude coordinate numbers such as

-110.3484, 44.2856
-110.3463, 44.2889
-110.3511, 44.2902

Latitude, longitude coordinates are normally in decimal degrees as above, while the coordinate numbers in projected files are most often meters in X and Y directions from some origin known to the projection. It is as if the green sheet in the illustration above were an enormous piece of graph paper on which the map is drawn “full size” and then measured off in meters.

In a well run GIS system the internal coordinates of projected maps may be hidden from the user because the GIS software will automatically translate the internal map drawing coordinates into Latitude/Longitude values on the fly. Manifold, for example, will show cursor position in a projected map view using Latitude and Longitude values. What is going on is that Manifold is automatically translating internal projected coordinates like 44030984,38403080 into the equivalent Longitude and Latitude values.

Conic Projections

The main problem with cylindrical projections is that they do a poor job of minimizing distortion except for very close to the line of tangency. They are a poor choice for mapping large countries (such as the US or Russia) that have great East-West extents.

A better choice for mapping such regions is a conic projection, which projects shapes from the Earth’s sphere onto a cone. Cones, of course, can be unrolled into a flat sheet without any deformation. Locations near the line where the cone is tangent to the Earth will be relatively free of distortion. By using taller cones we can move the line of tangency nearer to the Equator and by using fatter, more open cones we can move the line of tangency closer to the pole.

We can see the practical effect of a conic projection by considering a map of North America shown in the Latitude / Longitude projection. This is an “unprojection” that simply takes each coordinate in degrees and plots it using equal sized X and Y degrees at all locations:
The geographic cylindrical projection greatly overstates the size of northern regions.

Using a conic projection, we can transfer the shape of North America to the cone (in the region marked in red on the cone) and then unroll the cone to make a flat map. That flat map can then be used as "graph paper" to measure off coordinate locations with which we could build a flat, digital map.

The resulting flat map provides a much better impression of the true shape of North America. It is interesting to note that since most schoolchildren are taught geography from maps using cylindrical projections that greatly distort Northern regions, the average person thinks Alaska and Greenland are many times larger than they really are. The above conic projection uses a tangent line cutting through the "lower 48" US states and so optimizes their appearance while understating the apparent size of Alaska.
When both are viewed in Lambert Conformal Conic projection using parameters midway between the "lower 48" and Alaska and Alaska is moved over the "lower 48" US and rotated to preserve apparent meridian angles, it's clear that Alaska is very large, but not as large as is commonly thought.

**Azimuthal Projections**

Azimuthal projections show one hemisphere of the Earth at a time by projecting lines upward from the globe onto a flat disk tangent to the globe at one point.

By centering the disk over any particular point on the Earth, one can achieve a view of the Earth as it appears from space from high over that point. The Orthographic projection is the classic "view from space" azimuthal projection of the Earth.

**Projections and Projection Parameters**

Virtually all projections in common use fall into one of the above three categories. They are either cylindrical (regular or transverse), conic or azimuthal projections as customized by slightly different projection parameters. Projection parameters are options in how the projection is arranged.

For example, the Orthographic projection can be centered on any point on Earth by specifying the latitude and longitude of the desired central point. Conic projections may be customized by specifying the parallel of latitude at which the cone should be tangent.

Specifying a projection together with various optional parameters will drive the mathematical conversion of longitude, latitude degree coordinates into the numbers used within the projected coordinate system. When we encounter a computer file with projected data numbers such as…

44030976, 38403088
44030984, 38403080
...we will not be able to make geographic sense of these numbers unless we know in which projection with which optional parameters they are intended to be used.

Some GIS formats are "smart" and automatically save the projection parameters in use together with the data. During import of drawings from such formats, Manifold will fetch all necessary parameters from such "smart" formats automatically and will load the coordinates properties for that drawing with the correct parameters necessary to use the data.

When importing projected drawings from legacy GIS formats that do not save the projection information with the data we will need to know what projection and parameters should be used with that drawing. We will then have to enter this information manually into that drawing's coordinate properties so Manifold can use the data as intended.

Projections dialogs in Manifold are set up so they automatically present available options for the projection in use. Some specialized projections allow specification of an elaborate set of optional parameters.

**False Easting and False Northing**

Once a map is constructed using a given projection, the map is a flat surface. Distances on that flat surface may be measured as X and Y rectangular coordinates, with the X coordinate being the distance to the right of the vertical line passing through the origin or the center of a projection. A negative X coordinate represents distance to the left. In practice a false X or false easting is frequently added to all values of X to eliminate negative numbers.

Likewise, the Y rectangular coordinate is the distance above the horizontal line passing through the origin or center of a projection, with negative Y being the distance below. In practice, a false Y or false northing is frequently added to all values of Y to eliminate negative numbers.

The use of false easting and false northing is a relic of days when map projection computations were done by hand, so that computation with negative numbers was less convenient. In modern times we let computers do all the computational drudgework so false easting and northing are no longer essential. However, they continue to live on within projected digital maps created using older methods. Manifold allows use of false easting and false northing with many projections.

**A Historical Note on the Round Earth**

![Christopher Columbus](image)

**Christopher Columbus** by Sebastiano del Piombo
School children are often wrongly taught that Columbus sailed Westward to China to prove that the Earth is round. Once launched on his journey Columbus is often portrayed as heroically pressing on despite the opposition of his sailors, who feared their little fleet would fall off the edge of a flat Earth. That is almost the exact opposite of the truth.

Most educated people in Columbus's day knew the Earth was round. In fact, they not only knew the Earth was round they knew the size of the Earth as well. Almost everyone except Columbus accepted the estimate for the radius of the round Earth computed by Eratosthenes of Cyrene (276-195 B.C.). Eratosthenes figured the Earth's radius to be about 6267 kilometers, a figure remarkably close to the modern mean of about 6371 kilometers. In the 1490's educated people had known for over one thousand five hundred years the actual size of the round Earth. Since ancient days cartographers had even created projections to deal with the representation of a round earth on flat maps.

Even many uneducated people knew the Earth was round. Among uneducated people sailors especially believed the Earth to be round because of the frequent observation at sea that tall points such as mountains come into view above the horizon as the distance to an objective becomes closer. Many "round Earth" visual effects incompatible with a flat Earth are easily seen by the human eye at sea.

Columbus met much opposition at Court to his plan precisely because people knew the Earth was a very large sphere. The ships of Columbus's day were so slow that they could not be loaded with enough food and water to voyage directly to China westward from Europe. Without the then-unknown continents of North and South America to use as re-supply points the direct voyage would be so long that the crew would die before making landfall.

Columbus based his plans for his voyage on the argument that the Earth is smaller than it truly is. Educated people were unimpressed with what they regarded as his chain of wishful-thinking assumptions that "proved" Eratosthenes was wrong and that a Westward voyage was just barely feasible. When Columbus launched across the Atlantic his sailors were fearful that in the event his estimate of the Earth's size was wrong and everyone else was right they would expire of thirst and starvation.

As it turns out Eratosthenes was right and Columbus was wrong about the size of the Earth. Columbus simply had the good fortune of rediscovering a New World (it was first discovered and then forgotten by the Norsemen) before he and his crew died proving the true size of the round Earth. In all fairness it should be pointed out that despite his flawed belief in a small world Columbus was a master admiral of unparalleled skill, intelligence and personal courage. A failed and quarrelsome administrator on land, Columbus is indisputably one of the greatest leaders who ever took to sea. He is alone among the early voyagers in executing and surviving four successful voyages to the New World.

**See Also**

**Projections**
Coordinates

Objects in drawings are defined by the coordinates necessary to draw them. **Coordinates** are simply the numbers for each X and Y position that defines the object.

Consider a drawing that contains an area, a line and three points.

The objects are defined by the coordinate numbers that are used to draw them in a "connect the dots" fashion. The area, for example, is nothing more than a list of five coordinates in a computer file:

```
3 7
2 9
3 10
5 9
5 8
```

Through various clever means Manifold knows how to interpret the above list of coordinates as an area and how to draw the area knowing correctly what is the inside and what is the outside of the area.

By themselves the raw coordinate numbers describe only the relative position and shape of objects within the drawing. We don't know, for example, if the "3" in the area is supposed to be three centimeters or if it is three degrees of longitude. If the "3" is not degrees of longitude we don't know from the number by itself if it is supposed to represent an unprojected coordinate in a CAD drawing or if it is a number that has some meaning within some specific geographic projection.

The coordinate numbers that specify objects are given meaning (such as scale and geographic positioning) by the parameters specified in the drawing's Edit - Assign Projection dialog. Manifold files save this additional information along with the
coordinates needed to draw objects so that when a Manifold file is opened the coordinates properties are automatically loaded with the correct values. This is all done automatically so we never have to worry about internal coordinates.

If we work only within Manifold we do not need to know much about coordinates nor will we need to use the Edit - Assign Projection dialog. It is typically used only once when importing projected drawings from legacy GIS formats such as ESRI .shp or AutoCAD .dxf. It is not used when importing unprojected drawings nor is it normally used for drawings created within Manifold or imported from "smart" formats. Such drawings will automatically be created or imported with the correct settings for coordinates properties.

Projected drawings imported from older formats such as those used by some legacy GIS systems will at times require manual entry of their coordinates properties within the Edit - Assign Projection dialog. To use this dialog sensibly we must understand how coordinates work in general and how they are utilized within Manifold.

Experts may consult the Projections Quick Reference and the Projections topics.

If you are new to coordinate systems in GIS or would like a refresher on coordinate systems and projections, consult the Coordinates Tutorial, the Projections Tutorial, and the Coordinates in Projected Maps topics followed by a quick review of the Projections Quick Reference as a summary.
Coordinates Tutorial
This topic explains coordinates for users new to GIS. Experienced users may begin with the Projections Quick Reference topic.

Overview

Drawings and images come to us for use in Manifold from many different sources. Drawings come in many different sizes using different scales. Images can be everything from a photograph scanned on a desktop scanner at 1200 DPI, where one pixel is less than a thousandth of an inch in size, to a data set acquired by satellite where every pixel is supposed to represent a square kilometer of the Earth’s surface.

It's obvious that if we want to use different drawings or images together in a map we will often ask the system to enlarge or reduce them in some way so their scales match up. If we want to use drawings or images together in a map, we will also need some way of specifying the desired geographic position of the images so that Manifold can draw them correctly aligned together with drawings or other images in a map window.

In the illustration above showing Highway 101 in Palo Alto, we will have to not only match the sizes of the drawing and the image but also to place each in correct geographic context in a Map. This will allow us to add other drawings and images and see them correctly overlaid as well.

Coordinates in Drawings

Drawings are for the most part simply lists of the X,Y coordinates needed to draw the objects they contain. Let's consider a simple example to see how this works:
Imagine taking a sheet of paper and sketching the layout of a chair, a side table and a table with our computer setup and telephone.

If we drew this on a piece of graph paper with a fine grid, we could measure the X, Y coordinate of every corner point of every rectangle by counting the little boxes from the origin in the lower left corner of the paper.

For example, the lower left corner of the table is forty units in the X direction and 20 units in the Y direction so the coordinates of this spot are 40,20. The four coordinate pairs needed to draw the table rectangle might be:

40,20
80,20
80,50
40,50

We could measure off each and every coordinate location in our example drawing above and create a file containing the coordinates necessary to draw a plan of our office setup. Digital drawings in almost any format are simply vast collections of numbers like these that specify the location of every coordinate needed to create the drawing in a "connect the dots" fashion.

**Specifying Size**

The list of coordinates will allow us to reproduce the drawing in the future, but it does not tell us the size of the paper used or what each "unit" is intended to represent in real life. For example, if we originally used graph paper with a one millimeter mesh to measure coordinates and then one day re-created our drawing using graph paper that had a 1/8-th inch mesh the two drawings would not be the same size. Likewise, it makes a big difference if each unit is supposed to represent one inch or one centimeter. A table 30 inches deep is passable while one 30 millimeters deep is not so good.

Units of measure don’t matter if we simply want an abstract view of the arrangement but they matter a lot if we wanted to combine this drawing with a different drawing to compare the relative sizes of tables, or if we wanted to use the drawing to make measurements that we would apply in real life.
We could add some metadata ("data about the data") to the drawing by writing in a note: "one drawing unit equals one inch" to tell us the scale. This would allow us to make measurements in our drawing and to compare those measurements to real life. "Smart" drawing formats used to save digital drawings will save the intended scale together with the lists of coordinates. This allows programs that read such formats to automatically reckon the correct scale. Surprisingly many formats fail to take this simple step.

**Specifying Location**

Looking at our sample drawing we may wonder why the chair is drawn above the table and not the other way around. We can specify which way our desk arrangement faces in "real life" by assuming that North is always up in the drawing. To do this, we can assume that Y coordinates going in the positive direction are headed North and that positive X coordinates are headed East.

We could also add metadata that specify where the drawing is located on the Earth.

For example, we could attach a note that the origin in the lower left-hand corner is precisely located at the longitude/latitude location of -122.1635, 37.4452. A mapping system could then know where to place it on a map of the Earth. If we could layer the drawing into a map, we could zoom far in and see that the desk is located at the old Microsoft Silicon Valley ISV center in Palo Alto.

"Smart" formats used for drawings and maps use a variety of ways to specify where a drawing should be located. One way is to use a standard geographic projection for drawing coordinates and to embed a note on the projection and projection parameters used within the drawing format. See the Projections topic for details on projections.

When importing a drawing from a format that does not provide information on the desired geographic location, we can georegister the drawing to the desired location. See the Georegistration topic for information on how to do this.

**Coordinates in Manifold Drawings**

Every drawing in Manifold includes embedded information about scale and location that describes the intended use of the X,Y coordinates it contains. This information is attached to the drawing when it is created or imported and can be seen at any time by looking at the drawing’s coordinate properties. If a drawing is imported from some "smart" mapping format the required information will automatically be imported with the drawing.
Drawings created using degrees, radians and other angular units are assumed to be Earth geographic degrees and will be displayed with the drawing centered on a 0,0 origin at the intersection of the Prime Meridian and the Equator.

Unless some other projection is specified, drawings created using any linear units of measure (inches, meters, kilometers, etc) will be created using Orthographic projection with the lower left-hand corner, the origin, also centered at the intersection of the Prime Meridian and the Equator.

**CAD Usage of Drawings**

Computer Aided Design (CAD) packages such as AutoCAD are used to create blueprints and other drawings without tying those drawings to specific geographic locations. We can do this in Manifold, as well.

We may wish to create drawings without specifying where they are located on Earth. For example, we might wish to use Manifold to create generic drawings of proposed factory layouts. We will probably want to specify the scale of the drawing, but perhaps we will not want to specify the geographic location of the drawings since we don't know in which location they will be used. No problem!

All drawings in Manifold, even drawings that are intended as CAD documents using meters as a unit of measure, are created by default using the Orthographic ("view from space") projection using a 0,0 origin at the intersection of the Prime Meridian and the Equator. This convention does not get in the way of CAD usage while it does provide a simple means of georegistration CAD drawings in a geographic context should the need arise.

If we are using Manifold as a CAD editor, it really doesn’t matter where the system thinks is the default geographic location of the drawings. The geographic location of the 0,0 origin of the drawing remains a theoretical matter buried in the drawing properties as long as we are doing "pure" CAD. It doesn’t surface as a consideration unless we wish to place the drawing somewhere within a geographic map. At that point it becomes a handy default placement.

So long as we are doing "pure" CAD, the only thing that matters is that the scale is accurate and that all the drawings are based on the same common position of the X, Y origin. There is no deformation from projection factors either, since the drawings used in our CAD project share a common meaning for their coordinates.
Suppose we create our desktop drawing in a Manifold drawing view using one inch per unit. We've illustrated a red cross at the lower left corner to emphasize the idea that the drawing uses internal coordinates based on X,Y displacement from that origin. Manifold does not actually show red crosses at the coordinate origin of drawings.

If we look at this drawing in a map it appears the same; however, if we switch the cursor X,Y readout indicator in the status bar to read in Latitude/Longitude instead of in X/Y coordinates, we'll see that the system thinks the lower left corner is at the intersection of the Prime Meridian and the Equator.

Suppose we create another drawing to show the arrangement of cubicles in the office room in which our desktop lives:

If we open this drawing in Map view it too will appear with the lower left-hand corner georeferenced to our map origin.

If we opened both drawings together in the Map view we would see the above situation. The scale is correct, since our desk set is supposed to fit in one of the cubicles. Now that...
we are looking at two drawings at the same time we need to show where they are positioned in relation to each other. To do this we simply select the desk set drawing objects and move them to where they are supposed to be:

![Desk set drawing objects moved](image)

There, that's better! Now we show our desk set where it belongs, in the second cubicle over by the left-hand doorway. Doing the edit in this way changes the internal coordinates used to make the desk set drawing so that every time we open it the desk set will appear in the right location when overlaid on the cubicle plan.

**Notes:** We've added the drop shadows and drawing outlines in the illustrations above to make it more clear in this documentation what is going on. Manifold does not show drawings in maps with drop shadows.

**CAD Style Drawings in Maps**

Normally, if we wish to create drawings for use in geographic maps we would begin with some geographic basemap drawing as a template. If desired we can always take CAD-style drawings originally created in a non-geographic CAD environment and see them in a geographic setting if we choose.

We can do all the CAD style editing we wish in a map using multiple drawings as multiple layers without ever caring that the lower left corner happens to be georeferenced to a spot off the coast of Africa. However we can exploit this default georegistration when mixing up drawings originally intended for CAD settings with drawings that have geographic content.

To see this effect, we can simply insert a world basemap drawing into the Map view we are using to view our "CAD" drawings.

The first effect is that we wouldn't see our CAD drawings any more because they would be a tiny dot off the coast of Africa and too small too see when the whole world is in view. (A drawing the size of our desk set is invisible at the scale of the entire world!) However, we could zoom far into that point and, like magic, see our desktop drawings appear in the size of our office setup at exactly zero latitude and longitude, apparently floating in the ocean off Africa.

We can make the desk set more visible by editing coordinates properties of our desk set drawing so that one unit is equal to 100 kilometers instead of 1 inch:
Suddenly the drawing appears as if our desk is 8000 kilometers wide instead of 80 inches wide. If we view this drawing in a map view together with a world map, we would see the lower corner of the desk is now 4000 kilometers (40 units x 100 kilometers per unit) to the right of the origin. Note that we haven’t changed the data in the drawing (it still contains numbers like 40, 20, etc), we have simply changed the way we want the system to interpret that data.

Because the default georegistration for drawings is to the 0,0 world Longitude/Latitude origin, this convention is a useful way of doing “quick and dirty” georegistration. Through a process of zooming in, selecting objects in a drawing, zooming out and moving the objects we can move objects to any position we desired. The Georegistration topic describes faster and far easier ways of accomplishing georegistration in Manifold, but this “thought experiment” is nonetheless a useful way of understanding how coordinates work in drawings.

**Coordinates in Images**

Images consist of adjacent pixels arranged in rows. There are no "coordinates" within images except those that may be inferred from the size of the image and its placement at some geographic location, if desired.

Just as the raw numbers within a drawing do not tell us what size the drawing is unless we know what size (inch, meter, etc.) the units used represent, we don’t know what the intended size is of an image unless we know how big the pixels are supposed to be. If we have an image that is 600 pixels wide and 900 pixels high, we don’t know what the
The intended size of the image is unless we know the physical size represented by each pixel. We can specify what size the pixels are supposed to be using any one of several methods.

In the case of scanned or other photographic images, the size of the pixels is usually specified by the dots per inch (DPI) resolution at which the image was created. If it was scanned at 300 DPI, there are supposed to be 300 pixels per inch so that each pixel is 1/300 inch in size. An image that is 600 pixels wide and 900 pixels tall that was scanned at 300 DPI is intended to be two inches wide by three inches tall.

DPI is fine for small images that have many pixels per inch. For images representing very large sizes we will most often specify the size of each pixel in meters or kilometers. For example, the very best commercial space satellite photographs will display an area one meter square for each pixel. If we had an image from such a satellite that was 600 pixels wide by 900 pixels tall, it would represent an area that was 600 meters by 900 meters.

When creating or importing an image in Manifold, we will be prompted to tell the system what size this image is supposed to be, either by specifying DPI or by specifying the size of each pixel. By default, Manifold will import images in geographic image formats using whatever scale factor is specified for the image. For non-geographic formats, Manifold will import the image using the screen resolution (typically, about 72 DPI) for the image. We can change this property at any time to rescale the image without losing any accuracy.

Until the image is georeferenced in a more sophisticated way, Manifold places it geographically with the lower left hand corner at the 0,0 origin at the intersection of the Prime Meridian and the Equator. Manifold's internal accuracy is so high that a 1200 DPI image can use the same system of 0,0 origin as drawings do by default and still be able to assign a specific lat/lon coordinate to each pixel, even though different pixels are only 1/1200 inch apart.

It is slightly ludicrous to think of a photographic image (like the detail above) being built up pixel-by-pixel using latitude and longitude coordinates; however, because Manifold does this internally when an image is used in a map it gives us great flexibility. We can mix images and drawings in a "CAD style" environment and know that an image scanned at 600 DPI from a letter-sized or A4-sized paper sheet photo will look exactly the correct size when layered in a map that includes a drawing made on the same scale.

Note that whereas drawings are made up of coordinates, images are made of pixels that are coordinate-free: coordinates of pixels are simply implied by the announced size of the pixels, their arrangement adjacent to each other in rows and the location of the image overall. While it is possible to reproject a drawing to change the coordinate values without any significant loss of accuracy, it is not possible to reproject an image without adding or deleting pixels. See Projections and Images for visual examples of this effect.

For this reason, to maintain accuracy we would not normally re-project an image that represents a raster data set because re-projecting the image changes the actual data by virtue of the interpolation necessary to change the shape of the image data into a new projection. If we are using the image for purely visual effect, of course, or if the interpolation is acceptable to us for the intended use of the raster data then we should feel free to re-project the image to match the desired map's projection.

Additional Reading

Experts should scan the Projections Quick Reference topic.
If you are new to coordinate systems in GIS or would like a refresher on coordinate systems and projections, consult the Projections Tutorial and the Coordinates in Projected Maps topics followed by a quick review of the Projections Quick Reference topic.
Coordinates in Projected Maps
Before proceeding with this Topic, please read the Coordinates topic and Projections Tutorial topic first.

In the Coordinates topic we saw how a drawing can be created on graph paper with a sequence of X,Y coordinates measured from the origin showing how to draw every object in the drawing. Drawings are just computer files filled with sequences of coordinates that are used to make the drawing in a “connect the dots” fashion.

To create a “flat” digital map, we use the same approach we did in drawing a plan of our office desk set in the Coordinates topic.

Recall the illustration of a cylindrical projection from the Projections Tutorial topic. It showed how a small part of the surface of the Earth could be “projected” onto a section of cylinder, which could then be unrolled into a flat sheet. If we want to save the X,Y locations of points on our flat sheet we can now measure them as though the flat sheet were graph paper to create a diagram of geographic features, just as we created a diagram of our desk set on a sheet of graph paper.

The above illustration shows a key concept that often proves confusing to GIS newcomers: although “unprojected” data about locations on the Earth are specified in degrees, all projected maps specify the coordinates of the objects on them with X,Y coordinates that are numbers representing meters, feet or other linear measures. These coordinates are computed relative to some origin established by the projection in use.

Computer files that contain projected maps therefore contain coordinates like

44030976, 38403088
44030984, 38403080
...and not longitude,latitude coordinate numbers such as

-110.3484, 44.2856
-110.3463, 44.2889
-110.3511, 44.2902

Latitude,longitude coordinates are normally in decimal degrees as shown above, while the coordinate numbers in projected files are most often meters in X and Y directions from some origin known to the projection. It is as if the green sheet in the illustration above were an enormous piece of graph paper on which the map is drawn "full size" and then measured off in meters.

In a well run GIS system the internal coordinates of projected maps may be hidden from the user because the GIS software will automatically translate the internal map drawing coordinates into Latitude/Longitude values on the fly. Manifold, for example, will show cursor position in a projected map view using Latitude and Longitude values. What is going on is that Manifold is automatically translating internal projected coordinates like 44030984,38403080 into the equivalent Longitude and Latitude values. Manifold allows "toggling" the status XY indicator between native units and lat/lon.

New GIS users are often fooled by the system's "translation on the fly" into thinking that their projected maps still contain coordinates in latitude/longitude degree form. This is not the case, since once a drawing is projected out of Latitude / Longitude into a different projection it will be in meters or other linear measure even though many Manifold dialogs will show the contents in degrees for the convenience of the user.

**Projected Coordinates and Projection Parameters**

The particular set of coordinate numbers in a projected file will make sense only if used within the projection parameters and projection within which they are intended to be interpreted. The projection parameters in use for a drawing may be seen at any time by viewing its coordinate properties. Click on the drawing in the project view to highlight it and then choose the Edit - Assign Projection dialog to open the properties dialog for that component. The coordinate properties tell Manifold how to interpret the data in that drawing. If these properties are changed, they simply change how Manifold uses the existing data. Changes in this dialog do not change the data itself.

For example, if we imported the following sequence of coordinate numbers...

44030976, 38403088
44030984, 38403080
44030900, 38403077

...from an ESRI "shapefile" they would make no sense if the coordinates properties were set to use Latitude / Longitude with coordinates interpreted as degrees. If we knew these coordinates were created for use with some particular UTM zone projection parameters we could open the Edit - Assign Projection dialog and enter those UTM zone parameters. The drawing would then make sense within Manifold.

Note that there is nothing about the raw numbers that says how they should be interpreted. "Smart" GIS formats will always save the required projection information with the file containing the coordinate numbers. When importing drawings from such formats
Manifold can automatically grab the required projection information and use it for that drawing's coordinate properties.

Legacy GIS formats do not save the necessary projection information with the files containing projected coordinates. This poses a big problem if one acquires a file full of projected coordinates without knowing what projection was used. ESRI's "shapefile" format is a classic legacy format that does not include projection information. It's best to use this and other legacy formats only for unprojected maps that require no elaborate projection parameters.

**Changing Projections**

Changing projections in a Manifold map is easy: we simply use the Edit - Assign Projection dialog to specify whatever projection we want and Manifold will automatically display all drawings and images in that map within that projection. There's no change to the native projections used by the drawings and the images since map windows in Manifold re-project their contents on the fly.

Because drawings are normally seen through map view within maps, it is not normally necessary to reproject them from whatever projection they were in when imported. Specifying the projection used by a map in map view does not change any data - it simply changes the way the data is seen in that particular map view. If desired, we can see the same drawings simultaneously through different map views using different projections in each map view.

The only time we might need to reproject a drawing is to speed up map view. Map view works much faster if the drawings it contains use the same projection requested of map view. This is especially true if many, large drawings and images are used in the map.

The X,Y coordinates in a projected drawing make sense only within the "grid" coordinate system used by that particular set of projection parameters. If we wish to change the projection within which the drawing exists we need to re-compute the numbers into their equivalents in the new projection.

Manifold does this via the Edit - Change Projection dialog. Specifying a projection with any optional parameters in this dialog will cause the coordinate numbers within the file to be changed into their equivalents in the new projection. The **Edit - Assign Projection** dialog will also be updated with the new interpretation.
Projections and Images

Images in Manifold may be projected as well as drawings. Images may be projected on the fly within maps or they may be re-projected within their native coordinate systems. Images are also re-projected automatically when they are georegistered to match a reference map, drawing or image.

Images have an implied coordinate system given by the row and column arrangement of their pixels. If an image is imported from a geographically aware image format such as GeoTIFF, it is automatically georeferenced to the correct location on Earth in geographic coordinates.

If an image is imported into Manifold from a geographically mute format such as .bmp or .jpeg, it is projected using Orthographic projection and placed with its lower left hand corner at the intersection of the Prime Meridian and the Equator (0 latitude and 0 longitude). This default georegistration decision together with size information about each pixel makes a specific choice as to where that image is located, what size it is and thus where the "center" of each pixel is located.

Just like a drawing, the coordinate properties currently in use for an image may always be seen in the Edit - Assign Projection dialog for that image. Changing any settings in this dialog does not change any pixels within the drawing - it simply changes how Manifold interprets the pixel data that already exist. To change an image's projection, we need to use the Edit - Change Projection dialog.

Although projection in images and drawings is conceptually similar, we tend to work differently with images and projections than we do with drawings and projections.

- Images used for visual effects are often re-projected to achieve greater speed in a map view. Images usually contain many more pixels than drawings contain objects. The cost of re-projecting an image on the fly in a map window is therefore often much higher than the cost of re-projecting a drawing on the fly.
- Images are usually highly localized views shot from directly overhead or very nearly so. They thus may be treated as already being in Orthographic projection. In contrast, most drawings or imported maps are in Latitude / Longitude coordinates and are not in any specific geographic projection.
- For geographic use, drawings are most frequently imported from formats that provide a known geographic location. Images are most frequently imported from formats that provide no geographic context. Images therefore will usually need to be georeferenced.
- Images are often slightly off Orthographic projection, or they are shot at angles that result in a Tilted Perspective or Space Oblique projected view in the image; however, it is very rare that we have actual numeric parameters corresponding to a particular image. Images will therefore often need to be adjusted through irregular geometric or numeric methods to fit a known good map. Drawings, in contrast, usually are transformed only through specific formulae from one known projection into another known projection.
- Any re-projection of an image will add or subtract pixels whereas re-projecting a drawing will neither add nor subtract objects. For this reason, images that present raster data where the value of each pixel represents important data will not usually be re-projected.

Georegistration

Georegistration is the process of moving, re-sizing and re-projecting an image to match the geographic region it is supposed to represent. This is done using control points to
match features in the image to locations in some reference map, drawing or image that is already in correct geographic context. This isn't necessary if we receive an image in a geographically aware format that automatically results in import into correct georegistration.

If we intend to use the image within a particular map it is wise to georegister the image using that map or using a drawing that employs the same projection used in that map. Doing so will georegister the image using same projection used by the map. This will eliminate the need to re-project the image "on the fly" when it is used within the map and will improve map display performance.

Once an image is georegistered to a known good geographic location and projection, we can use it like any other geographically accurate component.

**Visual Images and Raster Data Images**

Image formats are often used to deliver raster data such as terrain elevation or remote sensing information where the specific arrangement and value of pixels is very important. Any changes to such data sets (such as by interpolation required to change their shape into a desired projection) will add and subtract data not in the original data set.

Other types of image data are basically photographic images intended for human perception as visual images. In such cases, how the image looks to the human eye is more important than whether individual pixels are added or subtracted or otherwise changed. In many cases an overall change to the data such as improving contrast will improve the visual perception of the image.

Re-projecting an image invariably "warps" it in a way that adds or subtracts pixels. For visual images this is no problem. However, for images that represent specific raster data it may be unacceptable to change the data even if a very clever interpolation is used.

Within Manifold, we will therefore often re-project visual images without a second thought while taking care not to re-project certain types of raster data. Re-projecting photographic images into the same projection as a map view will allow the map view to operate rapidly. In the case of raster data set images we can always rely on the map view to re-project the image on the fly should it be necessary to show the raster data in a projected view. This will be slower but we will know the data has been unchanged.

To preserve the data accuracy of raster data images, they must be georegistered manually by altering their coordinate properties and not by re-projection through the georegistration process. To preserve unmodified data accuracy, raster data images must already be in some "natural" geographic projection such as Latitude / Longitude or be so close to a projection that no re-projection is necessary. Manual georegistration of such images requires opening the Edit - Assign Projection dialog and altering parameters there so the image is correctly interpreted.

**Why Re-projecting an Image Adds or Deletes Pixels**

All images imported into Manifold are rectangular because all raster image formats by definition have pixels arranged in rows and columns. Image windows show images as rectangles in the "native" coordinates implied by the row and column arrangement of pixels.
For example, images shot from oblique space vehicle trajectories and published as "north up" images will be shown in a rectangular format, usually with black pixels padding the extra regions needed to make the image a perfect rectangle. The image above is a Landsat 7 shot of the region near San Francisco Bay.

Although images are received as rectangles, re-projecting an image invariably changes its shape and leads to interpolation that changes the data. Re-projecting an image is, in effect, creating a new image.

Consider an image in its original rectangular format. All pixels are in locations adjacent to each other.

If we now re-project the image into some new projection (such as tilted perspective) we can imagine that the implied pixel locations will be pushed together in some regions and pulled apart in others. When shown in a map window within some projection the image will thus no longer be rectangular in shape. We should keep in mind this is only a thought experiment since images always consist of pixels adjacent to each other. Although we
can imagine that pixel locations are pushed together or pulled apart in real life an image always consists of a rectangular array of pixels.

When re-projecting an image, Manifold must take the implied locations of the pixels in the new form and transfer them to a new rectangular array of pixels. The color for each pixel in the new array is determined by sampling those color values at the implied projected points within it.

When georegistering or projecting images the **Size** parameters specify what size the new image is supposed to be. Using fewer pixels in height and width will result in re-sampling that averages down the image. Using a higher number of pixels in height and width will provide finer sampling. By default, Manifold tries to guess at a reasonable **Size** setting that will preserve the approximate overall size of the image when re-projection does not result in radically different shapes.

Note that increasing the number of pixels in height and width of the re-projected image can not increase detail or improve resolution beyond that in the original image. Increasing the number of pixels in height and width beyond the size of the original image simply re-samples the same "big" original pixel over and over within adjacent new pixels.
The result of re-projection is an image that has visible pixels in the appropriate locations and otherwise has invisible pixels. Using invisible pixels is more flexible than always using black pixels to "pad" the uneven edges of images to save them in rectangles. If desired, the Paint Bucket tool can always be used to pour black color into the invisible pixels.

Note that unlike re-projecting images, re-projecting vector drawings does not involve additions to or deletions from the data set. Even though re-projecting a vector drawing changes the shapes of objects it does not add or subtract objects from the drawing nor does it add or subtract from the number of coordinate pairs used to draw each object. The shapes of all objects in drawings are defined by the coordinates used to draw those objects. Re-projecting a drawing simply changes the coordinates into different numbers but it neither adds to nor subtracts from the coordinate pairs used to draw objects.

Note: The images above are simulations and are not actual screen shots because Manifold shows images in projected views too smoothly for a real screen shot to show the effects above.

Re-projecting for Speed in Maps

When an image is displayed in a map view using a projection other than the native image projection, the map view will automatically interpolate on the fly to add and delete pixels as necessary. This can be a time-consuming process for large images and will slow down map view in such cases. Note that even though the map window is showing the image in a new projection it is not changing the actual image data. It is simply re-computing on the fly how the image would appear in the new projection.

If we get tired of a slow map window when large images must be re-projected we can open the image in an image window and permanently re-project it to the new projection using the Edit - Change Projection dialog. When the native projection used by an image is changed the image will be re-computed and pixels will be added or subtracted as necessary for the new shape. Manifold will "pad" the new, non-rectangular shape of the image with invisible pixels to maintain a rectangular format. This will speed up map view if the new projection is exactly the same as that used in the map. This is a permanent change in the actual image data.

Mystery Projections

We will often encounter images that are close to, but not quite, a direct overhead view of a geographic scene where we do not know the exact circumstances under which the image was created. Such images are effectively in a Tilted Perspective projection where we don't know the parameters - a mystery projection, as it were.
We can deal with this situation by using georegistration to warp the image with control points to make it fit into some known good drawing. Instead of using ellipsoidal trigonometric formulae to achieve a perfect mathematical transformation between two projection coordinate systems as is done with projections in drawings, this method simply uses "best fit" algorithms to distort the image so that it is a good fit as directed by the control points.

**Tech Tip**

When an image is used in a map, for speed of operation the map usually is specified to have the same projection as the image. This is automatically accomplished when creating the map by using the image as the first component in the map. Sometimes the map is created first, an image is added to the map and then we would like to re-project the image so it has the same projection as the map. This is easily accomplished by right clicking on the image's layer tab and choosing **Project to Map**. This will re-project the image to use the same projection as the map.

**For advanced users:** Expert users might notice that sometimes doing a **Project to Map** will result in an image that looks slightly different when seen in its own component window than it appears when seen in a map. The reason for this is that **Project to Map** tries to preserve the original resolution of the image by modifying the values of the local scale parameters, in some cases choosing values for X and Y that are different for a better match to the original resolution. (To be exactly precise, **Project to Map** also modifies the values of local offsets as well, although this will not change the appearance).

As a result of the modifications in local scale parameters there can occur situations where an image component window and a map window showing the same image will not look exactly the same after re-projecting an image component to a map using the **Project to Map** command.

To make both windows look exactly the same, follow this procedure:

1. Open the image window and choose **Edit - Change Projection**.
2. Select the coordinate system to be the same as the map.
3. Press the **Suggest** button to let the system compute optimal values for local scales and offsets. These are the same values that would be used by the **Project to Map** command, were it to be used.
4. Modify the values of the local scale parameters so that they are equal to each other. There is no need to modify the values of local offset parameters.
5. Press **OK** to re-project the image.

**See Also**

**Changing a Component's Projection**
Changing a Component’s Projection

To permanently change a component's projection, choose the Edit - Change Projection command. This dialog permanently changes the projection used for a drawing or image. With drawings, it changes the coordinate numbers within the drawing to their equivalents in the new projection, and then updates Manifold's coordinate properties for that drawing. With images or surfaces, it will re-compute the image by interpolating to add or delete pixels as necessary to match the image shape to the required projection.

**Projection Dialog**

The dialog opens with a display of the current projection in use. Changing values in this dialog and then pressing apply will cause Manifold to re-compute the coordinate numbers within the component into their equivalents in the new projection. It is a permanent way of casting the data that define the drawing, image, surface or labels component into a new projection.

The dialog includes these controls:

- **Load from File** - Load projection information from a file using XML, Golden Software GSR or ESRI PRJ files.
- **Save to File** - Save projection information to a file using XML, Golden Software GSR or ESRI PRJ files.
- **Recent settings** - Choose a projection setting to reuse from a list of recently-used projections employed in this Manifold session.
- **Load from Component** - Load projection settings from another component in this project.

**Projection**

A tree diagram providing numerous projections within various types of projections.

**Datum**

A list of standard datums that incorporate various Earth ellipsoids and standard offsets.

**Parameters**

One or more optional parameter settings, if used by this particular projection. Double-click into the values boxes to change them. Press Enter after changing a value.

**Units box**

Choose the unit of measure. All "unprojected" maps are in Degrees, Arc Minutes or Arc
Seconds. Most projected maps are in meters.

**Local offset**  
Shifted version of the external coordinate system peculiar to this component. Specifies shift in X (longitude) and Y (latitude) direction in the given units. With images or surfaces, the Local offset stores the offset to the left-bottom corner of the left-bottom pixel of the image or surface.

**Local scale**  
Scaled version of the external coordinate system peculiar to this component. Specifies scale factor to be applied to external coordinate system.

**Scale correction**  
Controls scale of the external coordinate system. For example, 0.9996 for UTM.

**False easting/northing**  
Shift of the external coordinate system. For example, 500,000 meters for UTM. Some projections (such as UTM) add values to the "true" X and Y coordinates so that all coordinates stored are positive numbers. This avoids the use of the minus sign in primitive formats. The value added to X is the Easting and that to the Y is the Northing.

**Adjust for units**  
Automatically adjust easting/northing values when units of measure are changed. Not checked by default.

**Clip coordinates**  
Enabled for projections like Orthographic: clip those parts of objects that extend past the projection extent.

**Autosuggest local values**  
Enabled for images and surfaces: Automatically adjusts the values of local scale and local offset parameters after we change the coordinate system or alter one of the parameters in the parameter list. Checking Preserve local values overrides this box and keeps local values as they are.

**Preserve local values**  
Normally, choosing a projection from the projections pane will reset the values of Local offset, Local scale and units to the preset values for that projection. Checking this box will preserve any existing values in these local fields even when a new projection is selected. It is not checked by default in the Edit - Change Projection dialog because usually when re-projecting a component we want to use the preset local values associated with a new projection.

**Suggest**  
Suggest values for projection parameters based on the view in the component window. Used with projections with user-specifiable parameters.

This dialog should be a relatively infrequently used dialog since drawings and images are normally viewed as layers within maps. Because maps can show drawings and images in
whatever projection is requested of the map by re-computing the necessary views on the fly, it is rarely necessary to change the native projection of a drawing or an image. We would normally do so only to improve the speed of map windows.

Maps can work faster if the drawings and images they contain as layers use the same projection that is requested of the map. If we always use a particular projection within a map, it will be worth it to re-project the constituent drawings and images of that map into that particular projection as well. To permanently re-project a drawing or image, use the Edit - Change Projection dialog. In order for maps to run faster when the constituent drawings or images are in the same projection, they must use that same projection in all parameters. It does not help if both the map and its constituent drawings use Lambert Conformal Conic projection but with different center latitudes and longitudes.

Projection Toolbar Buttons

We may want to re-use the same projection information with different components. For example, suppose we have many shapefiles covering the same region of interest that contain projected information requiring the use of the Edit - Assign Projection dialog after import. Since all of the files will require the same projection information, after importing one file and manually specifying the projection settings in the Edit - Assign Projection dialog we can save the projection information using the Save to File toolbar button. With subsequent files we can then use the Load from File button to load the settings and thus avoid manual re-entry of repetitive information.

Another time saver is to load projection information from an existing component using the Load from Component button. For example, we may want to re-project all files into a certain standard projection used by a given component and would like to be sure that all settings are identical. In that case, we open each component to be re-projected, launch the Edit - Change Projection dialog and use Load from Component to load the desired settings from the component we wish to use as the standard.

Sizing Images and Surfaces

When choosing Edit - Change Projection to re-project an image or surface the image size will often change slightly in size. The actual size of the resulting image or surface in pixels will be determined by the Local Scale parameters, which set by implication the size of each pixel. Choosing inappropriate scale parameters can result in absurdly large images or surfaces. Attempting to create a very large image or surface in this way will cause Manifold to raise a confirmation dialog.

Discussion

To change the native projection of a drawing, image, surface or labels component use the Edit - Change Projection dialog.

Very Important: Do not use the Edit - Assign Projection dialog in an attempt to change the native projection of a component. This dialog is used to complete the import of a projected component from a format that does not correctly save projection information. Instead, use the Edit - Change Projection dialog to re-project a component.

Edit - Change Projection is virtually identical to the Edit - Assign Projection dialog because the two dialogs deal with similar information. However, the two are very different in function. The Edit - Assign Projection dialog the system how to interpret existing data without modifying the data. The Edit - Change Projection dialog changes the data to fit a required interpretation.
The **Edit - Assign Projection** dialog is used to make changes on a "one time" basis only when an import from a legacy format requires manual entry of the correct projection parameters. Changing parameters in the **Edit - Assign Projection** dialog changes only the interpretation of the existing data and not the coordinate data.

The **Edit Projection** dialog, in contrast, changes both the actual coordinate numbers as well as their interpretation. It is used every time one desires to re-project a drawing or image into a new projection.

Maps are a special case because they are viewports that show their contents in whatever temporary projection is desired. They don't change their contents, they simply show them in a different form. The projection used by a map to display its contents is set in the **Edit - Assign Projection** dialog.

### Local and External Coordinate Systems

The offset, scale, scale correction and easting/northing factors are intended for expert use. These parameters are scale and offset values for use with either the local coordinate system or the external, projection coordinate system.

In a nutshell, the external coordinate system is a coordinate system specified by the projection and projection parameters as described in the usual treatises on projection systems. The internal or **local** coordinate system is the coordinate system peculiar to the component, which may be a scaled and shifted version of the external coordinate system. Local systems occur when working with images imported from certain formats and possibly with drawings as well.

The **Local scale** and **Local offset** values expose the internal values used by a component to control the scale and shift of the internal coordinate system. These are given in the projection's units of measure displayed in the units box.

The **Scale correction** and **False easting/northing** parameters control the scale and shift of the external coordinate system. "False easting/northing" is used instead of "offset" for historical reasons. These factors are commonly met in projection systems like UTM. **Scale correction** values are given in dimensionless units (such as 0.9996 for UTM) and **False easting/northing** values are given in degrees or meters depending on type of the coordinate system (such as 500,000 meters for UTM).

**Example:** When importing a Manifold System Release 4.50 map that uses a **Unit2Degree** factor of 10,000 (the default), the **Local scale** will be 0.0001 for both X and Y. This means that one native unit within the imported drawing equals 0.0001 degree. Release 5.00 still uses full 1:250-millionth of a meter precision, but it imports the unit numbers from a 4.50 map knowing that the units used in that drawing had a scale factor of 0.0001.

**Example:** Increasing the **Local scale** in X of an image or drawing used as a layer in a map will make it look wider. Increasing the **Scale correction** in X will make it look narrower.

**Example:** Increasing the **Local offset** in both X and Y of an image or drawing used as a layer in a map will move the component towards the upper-right. Increasing the **False easting/northing** in both X and Y of an image or drawing used as a layer in a map will move the component towards the bottom-left (hence, the word "false").

**Local scale** and **Local offset** are applied to internal coordinates in a mirror image, inverse way as compared to how **Scale correction** and **False easting/northing** are applied to external coordinates within a projection system. In general, one can achieve exactly the
same scale or offset effect by applying inverse values to either the local or the external scale/offset pairs.

The choice of which to use depends on the context of the activity. **Local scale** and **Local offset** values are better suited for manual georegistration while **Scale correction** and **False easting/northing** values are better for the professional use of projections.

**Native Coordinates**

Manual edits of coordinates, scripting and use of intrinsic fields in tables may involve use of **native coordinates**. Native coordinates or native units are numbers stored within the drawing that Manifold uses internally as positional coordinates. These numbers are shown in the Object Coordinates dialog that pops up if you right-click an object and choose **Coordinates**. They are also used in the X (I) and Y (I) intrinsic field columns.

Projected coordinates are native coordinates adjusted with the local scale and local offset parameters of the coordinate system (projection) in use. In many cases, projected coordinates are the same as native coordinates (because of the local scale and offset values) but sometimes they are different.

The relationship between projected coordinates, native coordinates and the local scale and local offset parameters may be expressed in pseudo-code as:

\[
\text{ProjectedX} = \text{NativeX} \times \text{LocalScaleX} + \text{LocalOffsetX} \\
\text{ProjectedY} = \text{NativeY} \times \text{LocalScaleY} + \text{LocalOffsetY}
\]

**Adjust for units Option**

When manually changing units of measure the false easting and northing values will not be changed correspondingly. Checking the **Adjust for units** box will force a recomputation of the easting and northing values into the corresponding values for the new units of measure. For example, to change from Feet to Meters as units of measure and to change easting / northing accordingly:

- Verify the **Adjust for units** option is off,
- Set the units box to **Foot**,
- Turn on the **Adjust for units** option,
- Set the units box to **Meter**.

**Differences between Assign Projection and Change Projection**

- **Assign Projection** appears both with map windows and with individual component windows. When used with map windows it changes the projection used in the map view without actually changing the data in the components. Map windows have dynamic projections that can show their contents in whatever projection is desired regardless of the native projections of what is inside them. When used with component windows, this dialog changes the projection assigned to be used to interpret the coordinate data within the component.

- **Change Projection** appears only with individual component windows such as drawing windows or image windows. This command changes the actual coordinate data in the component into a new coordinate system. It can only work correctly if the starting projection assigned to that component was correctly assigned, so before we can use **Change Projection** to change a projection we must use
Assign Projection to verify that the projection originally assigned was correctly assigned.

Additional Reading

Experts may jump directly to the Projections Quick Reference topic.

If you are new to coordinate systems in GIS or would like a refresher on coordinate systems and projections, consult the Coordinates Tutorial, the Projections Tutorial, and the Coordinates in Projected Maps topics followed by a quick review of the Projections Quick Reference topic for a summary.

See also the Projections and Images topic.
Segmentization

*Segmentization* is the use of redundant, extra coordinates to define very large straight-line shapes so that they may be projected correctly. This is an advanced topic required only when creating very large geographic objects that have straight-line features. This topic assumes the reader understands how coordinates are used to define objects and how coordinate systems are used in projections. Let's begin with a refresher on how objects are drawn within projected coordinated systems.

Recall that all objects in Manifold ultimately consist of straight-line segments between the coordinates that define those objects. Even smoothly curved lines will be seen at high magnification to consist of straight-line segments between coordinates that define changes in direction. When re-projecting a drawing the coordinate numbers are changed to the new locations for each coordinate pair and then objects are drawn in the usual way in a "connect-the-dots" fashion as straight-line segments between the coordinates.

Consider a map of the United States shown in *Latitude / Longitude* projection. Suppose we use the *Shapes toolbar* to create a sequence of straight lines between points as shown. The two coordinate pairs at its beginning and its end define each straight line. A single coordinate pair defines each point.

If we change the map's projection to *Lambert Conformal Conic* the position of all coordinates will change slightly as required by the new coordinate system. Because the straight lines are defined by only two coordinate pairs they appear in the projected map as straight lines as well. This is fine as an abstract idea but it is not an accurate rendition of how a straight line in a Latitude / Longitude projection should appear in a Lambert Conformal Conic projection. If a feature is straight within a Latitude / Longitude projection unless it is a Meridian it should be gently curved in a Conic projection.
Let us consider an even more blatant example: Suppose we use the Shapes toolbar to create a rectangular box as illustrated within a Latitude / Longitude projection. The box is created as a single line that runs from a corner that is the beginning of the line through three additional corners and then back to an end position that is at the same corner as the beginning. Within the drawing that contains this line it is defined by a list of five coordinate pairs. Each pair defines the beginning or the end of the line (two pairs) plus one coordinate pair for each of the three other corners.

If we change the map’s projection to Lambert Conformal Conic the box (in black) will appear as a rhomboid with straight lines at the top and bottom; however, this is not a correct visual representation. A correct projection of the shape shown in the first illustration would appear as the magenta figure with curved top and bottom line. The box is wrongly portrayed with a straight top and bottom because those parts of the line run in a straight-line segment between only two coordinate pairs.

If we would like the box to have a gently curved top and bottom as is called for by the projection we need to provide extra coordinate pairs that can be moved as part of the re-projection.

If we look at the upper left-hand corner of the box in a close up zoom we can imagine a red dot showing the location of the coordinate pair defining that corner. (The coordinate pair defining the upper right-hand corner of the box is out of view off to the right.)
There's no reason we could not define the upper part of the line as a series of coordinate pairs. For CAD purposes, these are redundant. For geographic purposes, each coordinate pair is another location in the connect-the-dots list defining the line that can be moved slightly to achieve a curve in a projected map.

In fact, that's how the magenta figure was created in the illustration, by taking a copy of the black box and applying the Segments command in the Transform Toolbar for drawings. This command replaces straight-line segments of lines with additional segments using redundant coordinate pairs. The projection transformation then moved each of these coordinate pairs slightly to achieve a smooth curve instead of one straight line. The zoomed-in screen shot above shows alternate segments colored in red. Approximately 100 segments were used to achieve smooth top and bottom curves.

**Note:** The Segmentize command does not create new line objects. It starts with one line object and ends up with one line object. However, the line object it creates uses many more coordinates than needed by the original object to define the same line. The illustration above was created by taking a single line, breaking it into multiple lines and then coloring alternate lines red and black to show where the original segments were located. Even though the illustration uses multiple lines to convey the idea of segmentation, the reader should not be misled by it into thinking that the segmentize operator creates multiple lines from a single line object.

**Use in Geographic Maps**

Most features in geographic maps that are big enough to suffer visible distortion in projected maps will consist of hundreds or thousands of coordinate pairs. When projected, there are so many coordinate pairs available to move slightly in the re-projection that the map will appear smoothly projected. The risk of unwarranted straight lines appearing from the use of too few coordinate pairs happens only in the case of straight-line features that stretch over entire countries or very large states and provinces. Such features are rare and occur only in places such as the American West.

Geographic maps prepared by professional organizations such as USGS already use numerous redundant coordinates to define the straight-line sequences stretching over great distances. For example, the straight line defining the border between the US and Canada and borders of large, rectangular Western states is coded within USGS maps using many redundant coordinates. That is why the straight-line state borders in the US background map in the illustrations above appear smoothly curved in Lambert Conformal Conic projection.
Lack of enough coordinate pairs becomes an issue only when we create our own drawings on a continental scale or when we add objects in a simple way to existing drawings. In such cases, after creating the drawing or objects we should apply the Segmentize command to split the objects up into numerous redundant segments. This will provide many coordinate pairs to use as "handles" when re-projecting the drawing.

There is no need to use Segmentize within CAD drawings since all such drawings (factory plans, local site blueprints, etc.) are normally so small in geographic extent they are unaffected by projection distortion as are large states or other geographic shapes on a continental scale.

**Available Commands**

Segmentization can be accomplished either by using the Segments operator in the Transform Toolbar or by using the Segmentize command in the Drawing menu. The Segments transform operator adds the specified number of segments, whereas the Segmentize command adds segments so that each segment is no greater than a given Distance parameter.

**See Also**

Segmentize
Segments

**Georegistration**

Georegistration is the process of adjusting one drawing or image to the geographic location of a "known good" reference drawing, image, surface or map. For brevity, this topic and other georegistration topics use images as examples. However, the same procedures apply when georegistering drawings or surfaces.

- The drawing, image, surface or map being used as a reference is called the reference component.
- The drawing, image or surface being adjusted is called the target component.

Control points are used georegister images, surfaces, drawings and labels components. During georegistration the target image will be re-projected to match the reference drawing.
using the control points as a guide. The image above shows conceptually how control points in a reference drawing are matched up with equivalent control points in a target image.

Control points are matched by their name. For example, the spot marked by a control point named "ControlPoint2" in the drawing will be matched to the location in the image marked by a control point that is also named "ControlPoint2." Control points may be called by any names so long as the names used for control points within the reference component or target component correspond to the same location. For example, we could use the name "San Mateo Bridge East" for control points in both the reference component and the target component. These two points with the same name will be matched during the georegistration process. The choice of names is up to you. Some users prefer long, descriptive names while others prefer short names.

Any extra control points within either the reference component or the target component will be ignored. If there is a "ControlPoint18" in the target component but no control point of that name in the reference component it will be ignored.

We use the Control Points pane to add new control points to a drawing, image or map. Control points may be added to maps so that they may be used as a reference component, but maps are not georegisterable as a target component since they are made up of images or drawings in layers.

To Georegister a Target Component to a Reference Component

1. Place as many control points as is possible in the target image or drawing to be registered. Control points should be as evenly distributed as possible.
2. Place control points in the reference component with names matching the equivalent control points in the target.
3. Click on the image or drawing to be registered to make it the active window and choose Register in the Control Points pane. Manifold will present a dialog asking which component to use as the reference component. Only components with matching control point names will be listed.
4. Chose the reference component desired. Manifold will match those control points in the reference component to the target component by name and will re-project the target component so that its control points are in the same locations as the reference component.

Control Points

<table>
<thead>
<tr>
<th>Name</th>
<th>X / Longitude</th>
<th>Y / Latitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control1</td>
<td>-122.480997</td>
<td>37.811238</td>
</tr>
<tr>
<td>Control2</td>
<td>-122.048519</td>
<td>37.467911</td>
</tr>
</tbody>
</table>

The Control Points pane always shows a list of control points defined for the current image or drawing together with their coordinates. When we click on a different window the Control Points pane will switch context to show any control points defined for the active window.

Control points may be added by clicking with the mouse into drawings, images or maps or they may be added by manual entry of X / Latitude and Y / Longitude coordinates. See the Control Points pane topic for details.

The Control Points pane is also used to rename control points or to change their location by changing their coordinates. We can click into the Name field to change the name from
the defaults, if desired. To change the coordinates of a control point we click into their X / Longitude or Y / Latitude fields.

**Tech Tip**

Uncheck the **Autoscroll window on edit or selection operations** option in Tools - Options when working with control points. Because using the **New Control Point** tool activates the mouse as soon as it leaves the Control Points pane, if this option is not deactivated the target window will begin to autoscroll as the mouse enters it.

**Georegistering Images**

The most frequent use of georegistration is to georegister images. Images, such as scanned paper maps, are often imported with the intent of tracing them to create new, custom drawings and maps. The first step in the tracing process is to georegister the image into the correct geographic location.

If the image includes features that also exist in a known good drawing we can use the drawing as a reference to georegister the image. For example, we may have scanned a paper map that shows property lines within a city boundary. We might already have a drawing that shows the city boundary but not the property lines. We can use the common feature of the city boundary to locate control points in both the reference drawing and the scanned image target. This procedure is discussed in the Georegistering an Image to a Drawing topic.

At times we need to georegister an image where we have no pre-existing drawing that has features in common with the image. To georegister such an image we need to identify features within the image for which we know the precise latitude and longitude coordinates. We can then use these features as control points based on the latitude and longitude coordinates they are supposed to have. For example, perhaps we have a small-scale paper map showing trails through a park where the park does not appear in any existing GIS drawing that we have. We would like to scan the map and georegister it so we can trace it. We can physically go out to different locations shown in the map and measure their precise coordinates with a handheld GPS device and then use these locations as control points. This technique is discussed in the Georegistering an Image to Known Coordinates topic.

Georegistering an image re-projects that image. Re-projecting an image invariably involves re-sampling the image by interpolating the image to a greater or lesser number of pixels. Images that are imported into Manifold from geographically-mute formats such as .jpeg or .gif do not have any scale associated with them. Georegistering such images into a geographic context will be the first time that the “size” of a pixel matters. By default, Manifold will attempt to preserve approximately the same number of pixels in height and width in the image. This can be adjusted if desired by changing the **Size** parameters in the georegistration dialog.

**Georegistering Drawings**

Drawings may also be georegistered. When we insert a new, blank drawing into a project it appears as a drawing in orthographic projection centered on the 0,0 origin of the Prime Meridian with the Equator. We can work with such drawings as if Manifold were a CAD editor to create blueprints, factory plans and other drawings where we do not care if they have or do not have a geographic context. We can also import pure CAD drawings from editors such as AutoCAD where the drawing has no geographic context.
We may wish to use such "CAD" drawings within geographic maps. For example, we may wish to place a CAD drawing of a real estate development within a geographic map of a town. To do this we can georegister the drawing to the desired location as is shown in the Coordinates Tutorial topic.

**Georegistering Surfaces**

Surfaces representing terrain elevation rarely need to be georegistered since most formats used in GIS to convey digital elevation can be imported as georegistered images. On occasion one encounters a format from which surfaces are imported that must be georegistered.

**Georegistration Dialog**

To georegister the active image or drawing, click on the **Register** button in the Control Points pane.

**Note:** Some options will be enabled only for certain methods or when sufficient control points have been defined.

- **Reference** The name of the drawing or image to be used as a source of control points to be used as reference points.
- **Method** One of the following algorithmic methods:
  - **Affine (triangulation)** - Also called geometric transforms, affine methods can georegister a target component to a reference component using fewer control points than required by the **Numeric** method. However, affine registration requires care in placement of control points.
  - **Affine (scale, shift, rotate)** - A fast method that is algorithmically equivalent to using **Numeric** with an **Order** of 1. This is the default method since it works with any number of control points.
  - **Simple (scale, shift)** - Match components using XY translation and re-scaling only. Works with any number of control points.
  - **Numeric (polynomial)** - Numeric matching uses numerical computation methods to transform one component to match another. A large number of control points are required for good matching but the method is algorithmically simple and fast. Available only when eight or more control points have been defined, thus allowing an order of 2 or greater.

  **Note:** Only those methods usable with the number of control points you have defined will be displayed. If you don’t see a method, add more control points.

- **Order** The level of mathematical sophistication applied. Higher orders result in better matches but take more time and more control
points. In numeric methods the highest order exponent used in the polynomial equations generated to transform the coordinate system of the target component. Enabled only for the **Numeric method**.

**Modify Coordinate System**

Enabled when the **Simple** method is selected. If checked (default) the coordinate system (projection) of the component being georegistered is converted to the coordinate system of the guiding component. If not checked, the coordinate system is not changed.

**Interpolate pixels**

Enabled when georegistering non-palette images or surfaces using the **Numeric method**. Creates a much smoother image when transforming the image into the new projection. Very computationally expensive: requires approximately twice the processing time to georegister an image if enabled. **Note**: Methods other than **Numeric** always interpolate pixels for non-palette images and surfaces and never interpolate pixels for palette images.

**Scale pixels equally in X and Y direction**

When this option is off, the user can specify both the width and the height of the resulting image or surface component. When the option is turned on, the user can only specify the width of the resulting component and the system will automatically compute the height. By default the option is turned on.

**Save error surface using**

Create a surface containing the root mean square error value for each location in the georegistered component. Choose the data type for the error number saved in the surface at each location. If checked, this option doubles the time required for georegistration. Enabled for the **Numeric method**. See the Error Surfaces topic.

**Size**

The transformed size of the image or surface in pixels. By default this will be set to some value that attempts to approximately preserve the size of the image or surface before georegistration. Enabled for images or surfaces.

The **Order** chosen will have a great impact on the number of control points required, especially when using the **numeric method**. For numeric georegistration, there usually must be at least four times the order number in control points, give or take a few. Thus, for numeric registration of order 4, there should be about 16 control points. This guideline is a minimum value. The **Order** value used can greatly effect the quality of the outcome. If the image is not georegistered well, try a different value of **Order**.

Certain arrangements of control points may require an even greater number of control points. Exception for the **Simple method**, the more control points the better. Images of scanned paper maps will often be registered with 35 to 70 control points in a production map environment when using **numeric** or somewhat fewer points for **affine registration**.
Affine registration works with fewer control points, as few as two or three in the case of **Affine (scale, shift, rotate)**. However, greater care must be taken when assigning affine control points. In general, control points should be placed for affine registration so that control points are drawn along the outer border of an imaginary shape without control points in the middle of the shape. For example, placing control points in a rectangular or rhomboidal arrangement is OK. Placing control points in a circle with several additional control points in the center of the circle is not OK.

**Affine** registration is a good choice when registering scanned images of maps that have a graticule grid of latitude and longitude lines shown. One can choose four intersections of latitude and longitude graticule lines that are near the four corners of the image to use as control points. Because fewer control points are required affine registration is often a better choice if we can place control points in a rectangular or rhomboidal arrangement.

**Numeric** registration may be a better choice when registering images to control points that are scattered throughout the image. When using numeric registration, evenly distributed control points will yield better results. This will, however, usually require many more control points than affine registration.

Either affine or numeric registration is a good choice when registering scanned images of maps that have a graticule grid of latitude and longitude lines shown if we are able to place many control points. One can choose a number of intersections of latitude and longitude graticule lines to rapidly mark many control points.

In either case, if control points are restricted to only part of the image it is quite likely that other parts of the image will not be georegistered well. When used in maps they will appear to be out of alignment with overlying drawing objects. It is critically important to use control points that are dispersed throughout the entire image. With the **numeric** method, the more control points that are used, the better the ultimate georegistration.

Given the labor of marking many control points the appeal of **Simple** registration is obvious. If an image is already in a North up overhead view and need only be resized and moved to be registered to a given target drawing this is a good choice. If the image is in Orthographic (the default when importing from geographically-unaware formats like .jpg), re-project the drawing into Orthographic and note the central latitude and longitude. Use **Edit - Assign Projection** with the image to specify the same central latitude and longitude. Then use **Simple** registration to georegister the image to the drawing.

**Example**

We begin with our **SanFran** sample Landsat image, which has been cropped to a smaller image centered upon San Francisco Bay.
The image has had six control points added that are reasonably dispersed near the edges of the image. This is a good pattern when the image is a near-overhead view and numeric registration is to be used. However, in "real life" we would use many more control points, probably over 30 scattered throughout the image. The image will be registered to control points in a drawing of hydrographic lines imported from a 1:100K-scale DLG downloaded from USGS.

After numeric registration, we can show the image in a map with the drawing lines show in yellow. Even though a small number of control points has been used registration is still very good.

**Using Maps as Reference Components**

If an image is being georegistered for display within a map it makes sense to use the map itself as a reference component, or to use a component that has the same projection as the map as the reference component. This will assure that the projection assigned to the image when it is georegistered will be exactly the same as that used by the map. The image will thus display faster within the map.
Maps are often a good choice for a reference component because they can show several layers such as roads and hydrography. Quite often a feature that is visible in an image may be identified with greater precision in a map when several layers are available to ascertain location.

For example, suppose we have an image like the one above where a bridge is visible. The example shows our SanFran sample image with the San Mateo Bridge crossing the South Bay.

The bridge provides a useful feature for use with control points, but if we work only with a hydro layer we will not see where the ends of the bridge could be marked.

Viewing both a roads layer and a hydro layer at the same time makes it easy to mark the ends of the bridge with control points. Note that the roads layer alone would not be any more useful than the hydro layer alone.
The resultant georegistration allows a reasonable match. The hydro outlines in the lower right of the image follow what USGS considers to be the shoreline in a region of tidal flats where a one foot (about a third of a meter) variation in water level changes the water’s edge by hundreds or thousands of feet.

**A Fast Arrangement**

Many users will set up their consoles so that the reference drawing and the target image appear side-by-side with the control points pane in the middle.

One can then zoom into approximately the same region in both map and image windows to add control points.

Workflow with this arrangement is a simple cycle: zoom into particular region in both the map and the image. Click on the image and add three or four control points. Click on the
map and add the analogous three or four control points. Click on the map and click the Back arrow to get back to the original view. Click on the image and click the Back arrow to get back to the original view. Repeat the cycle zooming into a different region to add control points.

Note that one can Zoom Box into a region, assign control points, and then click the Back arrow without losing the Zoom Box command. This sequence therefore is quick and easy to do. We can continue the zoom in / assign control points cycle until lots of control points have been assigned, and then georegister. With lots of control points the fastest method is usually Numeric.

Zooming into a particular region to place control points provides two main benefits:

- It allows us to place several control points in that region with zoomed-in accuracy.
- If the same region is zoomed in within the map window and the image window we are less likely to make a mistake when placing a control point because fewer points are involved.

The above advice, of course, assumes you have a reasonably high-resolution display with a reasonably large monitor so that there is enough room to have two reasonably useful windows open at once. 1024 x 768 is the absolute minimum, and really too small for serious georegistration work. Invest in a quality monitor or flat panel that's big enough to allow you to run at 1280 x 1024 resolution or greater.

Notes

Because control points can be imported from points that are in the drawing, a fast way of adding many control points is to add points into the drawing using Instant Data to specify the name of each point as it is clicked. We can then use these points as control points. See the Managing Control Points topic for essential methods that will save time and hassles.

Manifold will refuse to use control points that are placed on top of each other in the same image or drawing. The nature of control points is that they must refer to separate locations.

Most images that are overhead views of a subject can be georegistered to drawings in Orthographic projection using the Simple method with only two control points. This is a very fast match that is adequate for many GIS purposes.

The Save error surface as box option is not enabled for simple and affine methods.

Performance

Georegistering larger images can take substantial amounts of time. Depending on the size of image and the speed of the processor, large images can take hours to georegister. A very slow processor with an exceptionally large image might take days. The size of image, the speed of the processor and the method or other options selected can result in extensive variations in computation time required for georegistration. Checking the save error surface box will double the amount of time required.

For example, the Affine (triangulation) process is the most computationally intensive. Georegistering a 6500 x 7500 pixel grayscale image using the affine method on a slow machine with limited RAM, such as a 600Mhz PIII with 384MB of RAM, will take well over an hour. Georegistering the same image using the simple method (if it is a suitable
overhead image that can be treated as Orthographic) to an Orthographic drawing will require a few seconds on the same machine. Georegistering the same image using the numeric method (less computationally intensive) on a faster machine (1.1 Ghz Athlon with 512MB of RAM) will require 10 to 15 minutes. Depending on the hardware and method chosen the time required to georegister the very same image can range from seconds to over an hour.

Try smaller images, or try georegistering the image using a greatly reduced number of pixels from the defaults initially appearing in the Register box. Begin with very small images until you have some experience in realistic time required for your system. Use a lower Order with the Numeric method to speed it up. See the Performance Tips topic for suggestions on getting maximum performance out of your system.

Consider also the true size of an image. An image that is imported from a compressed file format, such as a .tif file, that uses only 50MB on disk may actually represent over nearly a gigabyte of pixels. Compute the size of the image by multiplying the X and Y dimensions in pixels times four for RGB images and times five for RGBA images. Grayscale images require much less space.

Finally, check to see if the image has invisible pixels. It is almost always best to crop the image to the desired area of interest right after importing it into Manifold, to make sure that it is not bordered with a swath of invisible pixels. Doing so avoids wasting computational time on georegistering or otherwise manipulating large numbers of invisible pixels that have no real role in the image.

**Technical Notes**

The Numeric method computes a set of coefficients for X and Y polynomials of the specified order minimizing the error at control point locations. The computed polynomials are then applied to transform the source data.

The Affine (triangulation) method utilizes a proprietary manifold.net algorithm that deploys not one, but many affine transforms.

Large images or surfaces can take a very long time to georegister when using Affine (triangulation) method or the Numeric method with higher Order. If desired, the georegistration process can be cancelled and then re-launched using a different method or lower Order.

Occasionally we might make a mistake in the placement or naming of control points that results in a bizarre and obviously wrong georegistration. In such cases, it is nice to know that Undo will work.

Images must be resized during some types of georegistration. Palette images will always be resized using the nearest neighbor method (no interpolation of colors) to guarantee that the georegistration process does not introduce any new colors.

**See Also**

- See the Manual Georegistration topic for discussion of georegistration by manual adjustment of projection properties.
- See the Error Surfaces topic for examples of usage, including visual examples of the effect of higher Order and control point location.
- For fine adjustment of registration, use the layer repositioning commands.
• The File - Print dialog used to print a component includes a **Control Points** option that may be used to print the control points in a component.

• The Georegister a Scanned Paper Map example topic shows a common georegistration task in step-by-step detail.
Georegistering an Image to a Drawing

Georegistering an image to a drawing is one of the most frequent tasks in GIS involving images. It's easy in Manifold:

1. Open the image to be georegistered. Place as many control points as possible in the image at locations that may be readily identified in the reference drawing.
2. Click open the reference drawing to which the image is to be registered.
3. Place control points in the drawing at locations corresponding to those used for control points in the image.
4. Click on the image to make it the active window and choose Register in the Control Points pane. Manifold will present a dialog asking which component to use as the reference component. The reference drawing used in steps 2 and 3 above will be listed in the list box of available reference components.
5. Chose the new drawing for use as the reference. Manifold will re-project the image to match the control points placed in the drawing.

See the Georegistration topic for additional details on how to use control points to georegister images and drawings.

For brevity, this topic and other georegistration topics use images as examples. However, the same procedures apply when georegistering drawings or surfaces.

**Tech Tip**

Before georegistering an image to a drawing, if the image is in Orthographic projection (the default for non-georegistered images), take a moment to re-project the drawing into Orthographic projection centered on the approximate central latitude and longitude of the drawing.

Images that are imported from non-geographic formats will import using the default Orthographic projection. In most cases, such images are overhead views photographed from aircraft or satellites. Their natural appearance, therefore, is already effectively the same as their appearance in Orthographic projection.

Drawings, on the other hand, are quite often imported from geographically aware formats and so will usually appear within some projected form. Due to the difficult of saving projection information in simple GIS formats like .shp, many drawings that are published on Internet are provided in unprojected Latitude / Longitude form. Such drawings may appear distorted as compared to a direct, Orthographic, overhead view, especially in higher latitudes.

If the distortion is great enough, the visual difference between the image and the drawing may make it more difficult to pick out features in common. By re-projecting the drawing into Orthographic projection we can given the drawing a visual appearance similar to the image. This will make it easier to pick out features that are common to both the drawing and the image.

**See Also**

Georegister a Scanned Paper Map
Georegistering an Image to Known Coordinates

At times we have an image that we would like to georegister where we do not have any known good components available for use as a reference component to which the image may be registered. If we have locations within the image for which we know the coordinates (say, as recorded using a GPS device taken to that location) we can nonetheless georegister that image.

Georegistering an Image to Coordinates

1. Open the image to be georegistered. From the Control Points pane use the **New Control Point** button to place at least three (and preferably many more) control points in the image at locations for which the coordinates are known.
2. Click on the project pane and insert a new drawing. This will be our target drawing.
3. Open the drawing. It will be blank. Re-project this drawing into Latitude / Longitude projection if it is not already in Latitude / Longitude projection.
4. In the Control Points pane use the **New Blank Control Point** button to add blank control points to the drawing that correspond to the control points placed in the image. For each control point, double click into the X / Longitude and Y / Latitude boxes to enter the known coordinates for the location of that control point.
5. After all of the control points are entered, re-project the drawing into whatever projection is desired as the end projection for the image.
6. Click on the image to make it the active window and choose **Register** in the Control Points pane. Manifold will present a dialog asking which component to use as the reference component. The new drawing created in step 3 above will be listed in the list box of available reference components.
7. Chose the new drawing for use as the reference. Manifold will georegister and re-project the image to match the control points placed in the drawing. After the georegistration process, the image will be in the same projection as the target drawing.

The above process is often made faster if we have used a GPS device to measure locations in the image and have stored each location as a waypoint in the GPS device. We can then download the waypoints into a drawing and use those points as our control points without the need to manually enter coordinates for each.

Another use of this process is to georegister a scanned image of a paper map. Paper maps often have graticules printed on them so we know the exact latitude and longitude coordinates at the intersection of the graticule lines. We can place control points at the intersections of the latitude and longitude lines in the image and then use these coordinates for the corresponding control points in the target drawing.

**Example**

Suppose we have a Landsat 7 image of the San Francisco Bay area. We would like to use this image to create new digital maps; however, we do not have any current digital maps to use as reference components. We do have latitude / longitude coordinates for two locations provided by friends with GPS devices who live in the Bay area. The first location is at the southern foot of the Golden Gate Bridge. The second location is at the south end of the Oakland International Airport.
We begin by using the Control Points pane’s **New Control Point** command to mark the known locations on the image with the mouse.

![Control Points](image)

We then insert a new drawing into the project. Make sure this new drawing is in **Latitude/Longitude** projection so that we can enter control points into it using latitude and longitude coordinates. To do this, first create a new drawing in the project pane, open the drawing and then use Edit - Change Projection to change the projection to **Latitude / Longitude**. It opens as a blank drawing.

<table>
<thead>
<tr>
<th>Name</th>
<th>X / Longitude</th>
<th>Y / Latitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>ControlPoint</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
</tbody>
</table>

In the Control Points pane we click **New Blank Control Point** to add a blank control point to the pane.

![Blank Drawing](image)

The new control point will appear in the blank drawing.

<table>
<thead>
<tr>
<th>Name</th>
<th>X / Longitude</th>
<th>Y / Latitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>ControlPoint</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
</tbody>
</table>

We double click into the X / Longitude field to change it.
We now type in the coordinate to use for Longitude.

When we push Enter the new coordinate is entered.

We can repeat this procedure with the Latitude coordinate.

Clicking New Blank Control Point we add another line to the Control Points pane.

We can manually add the coordinates for this new control point.

We can click Zoom to Fit in the drawing to show both control points in the drawing.

We can now click on the image to make it the active window and then press Register in the Control Points pane to register it in the usual way, using the new drawing in which we have created two control points. This example uses only two control points to save space. In "real life" we must use three or more control points. See the Georegistration topic and the Control Points pane topics for details in using these dialogs.

See the Georegister a Scanned Paper Map example for a detailed, step-by-step process to georegister an image to known coordinates.

Notes

As we edit the coordinates for control points manually by double clicking into the X / Longitude and Y / Latitude fields the control points being edited will jump about in the
drawing as their specified position moves about. Use **Zoom to Fit** to see the control points that have been added once manual editing has been completed.

Manifold maintains all coordinates to full precision. The default format for how many positions after the decimal point are shown in dialogs such as the Control Points pane is set in **Tools - Options**.

This topic is written showing the registration of an image as the target component. We can apply exactly the same procedure when registering a drawing (such as a CAD drawing of a factory plan) to known coordinates.

We can use tables to make lists of control points for use in georegistering images. When editing fields in tables, Manifold will automatically make sense of many different ways of entering latitude and longitude data into latitude and longitude type fields. This is extremely useful when entering coordinates for control points read off from paper maps that are marked using degrees, minutes and seconds notation. See the Create a Table and Add Records example for a step by step example of how one could add control points to a table using degrees, minutes and seconds format.

Suppose we have the GPS device in hand that was used to measure the known locations. We could have saved these in the GPS device as waypoints and then used the Manifold GPS Console to download the waypoints as points into a drawing. We could then use the **Load Points** command in the Control Points pane to load these points as control points. This would eliminate the need to enter coordinates by hand and thus also eliminate any possibility of typographic error when entering the coordinates.

For brevity, this topic and other georegistration topics use images as examples. However, the same procedures apply when georegistering drawings or surfaces.

See the Georegistration topic for additional details on how to use control points to georegister images and drawings.

Keep in mind that scanned images can be immense. An RGB image that is 12000 x 14000 pixels is over 800 megabytes in size. If you work with such large images you should have Windows 2000 or XP to avoid Windows bugs in 16 bit Windows versions such as Windows '95 or Windows '98. You should also have a **fast** processor and at least a gigabyte of RAM. Do not work with higher resolution images than is necessary.

**See Also**

Georegister a Scanned Paper Map
Error Surfaces

Georegistration is usually a tradeoff between a desire for high accuracy and the operator's patience or willingness to enter many control points or to tolerate long georegistration processing times required by using higher orders. When fewer control points are used the accuracy of georegistration declines. If desired, Manifold can compute a measure of accuracy of transformation when using the numeric method. The accuracy report is presented in the form of a surface where the value at each location in the surface provides a measure of georegistration accuracy at that location.

Checking the Save error surface as dialog box will create a surface in the same coordinate system and the same size as the component being registered. The value of the surface at each location reports the georegistration accuracy by reporting the RMS (root mean square) error at that location. The RMS values are computed by applying an inverse georegistration to the registered surface and then comparing the inverse result to the original component.

Beginning with an original component and then georegistering the component introduces some errors and then reversing the georegistration through an inverse transformation introduces additional errors. The component resulting from georegistration and then an inverse georegistration is therefore different from the original. It can be compared to the original using a root mean square comparison of original and transformed / re-transformed values for X and Y coordinates at each location.

The root mean square computation is reported in units of the source coordinate system. Values range from zero in regions where the transformation results in exact matches to some non-zero value in regions where the transformation and inverse transformation results in imperfect matches. The RMS error values are assembled into a surface called an error surface for convenient display of error values. The RMS error value in each pixel of the error surface is treated as the "height" of the surface at that pixel.

Error surfaces have their View - Display Options set to no shading and no palette. When opened in a surface window the error surface will have black tones in regions of low error transformations and lighter tones in regions of higher error transformations. High accuracy regions will be near control points and lower accuracy regions will be further away from control points.

If greater accuracy is desired we can then add more control points in regions of lighter tones and repeat the georegistration (save a copy of the original component so that the georegistration may be repeated using the original). Error surfaces created during georegistration will inherit control points.

Important: Computing an error surface requires not only the original georegistration transformation but also calculation of an inverse transformation. This requires additional computation and results in a longer georegistration process, approximately doubling the time required to georegister images and surfaces. Do not check the Save error surface as box unless you are willing to wait twice as long for the georegistration process.

Examples

Caution: This example uses some options that are not available in production Manifold releases. It is an artificial example cobbled up using a special version of Manifold created by manifold.net engineering. The special version was used to create illustrations of the effects of the Order parameter that will fit within the very small screen space available for Help illustrations.
The difference between this example and production Manifold releases is that this example uses the **Numeric** method with an **Order** of 1. That's not possible in production releases, where the minimum **Order** usable with the **Numeric** method is 2. However, using an **Order** of 1 provides a dramatic, obvious example when compared to an **Order** of 2 even though a relatively small number of control points (seven) has been used.

These examples use a portion of the **SanFran.jpg** sample image. The image was cropped and resized to a smaller number of pixels.

We have placed seven control points in the image that match seven control points in the **Bay_hydro** example drawing.

![Control Points](Sanfren.png)

Clicking the **Register** button in the control points pane launches the georegistration dialog. We choose the **Numeric** method, check **Save error surface as** and press OK. The **Order** is left at 1. [**Note**: Remember, this is possible only in the special version of Manifold used for this illustration. Production Manifold releases require a minimum **Order** of 2].
The image is georegistered to the **bay_hydro** drawing...

...and a new surface reporting errors is created.

If we open the error surface we see that by default it is seen in grayscale with no shading applied. Dark regions show lower errors and lighter regions show higher errors.

We can choose **View - Display Options** and assign a palette to the error surface. This also has the side effect of showing the minimum and maximum error values in the surface within the **Display Options** dialog.
Seen using the Spectrum palette the error surface shows finer visual detail of the distribution of errors. The surface is shown next to the georegistered image.

The error pattern seen in the above example is a very simple pattern that results from our use of a low order, 1, for the numeric georegistration. It’s fairly obvious why the Numeric method in production versions of Manifold requires an order of 2 or higher in order to avoid error surfaces like the above. If we were to repeat the georegistration using a higher order a more sophisticated georegistration would occur showing a more detailed error pattern.

To see this effect we can begin with a copy of the original image and use an Order of 2 in the Register dialog, as seen above.
The result is a slightly different appearance in the georegistered image as it is warped into registration with the latitude / longitude projection of the Bay_hydro drawing.

Opening the created error surface we see that it has a different pattern of grayscale tones.
Applying the **Spectrum** palette we can see that there is a region of high accuracy that correlates well to the placement of control points. The least accuracy occurs in the upper right corner of the image which is farthest from any control points.

Error surfaces are created using the same projection as the georegistered component. This allows us to overlay the error surface in a map with the georegistered component.

![Error Surface Image](image)

The illustration above shows the error surface in a map overlaid upon the georegistered **SanFran** image together with the **Bay_hydro** drawing. The opacity of the error surface has been reduced using Layer Opacity so the **SanFran** image may be partially seen through the error surface. This allows direct comparison between the accuracy implied by the error surface and the georegistered image.

**Technical Discussion**

As mentioned earlier, the above illustrations were created using a specially engineered version of Manifold that allows use of an **Order** of 1. Production versions do not allow this. Why was a special version created? There are two reasons why:

First, although production Manifold versions could use the **Affine** (scale, shift, rotate) method that is algorithmically equivalent to using **Numeric** with an **Order** of 1, only the **Numeric** method allows creation of an error surface since the **Affine** method is not exactly the same as the **Numeric** method.

Second, although it would have been possible to show a comparison between using an **Order** of 2 with an **Order** of 3. However, in order to use an **Order** of 3 there would have had to be many more control points in the image, so many that the relatively small sizes of images used for illustrations would have appeared very cluttered. By using a small number of control points any relationships between the pattern of control points and any resulting pattern in the error surface is more easily visible.

**See Also**

Georegister a Scanned Paper Map
Managing Control Points

Use the Control Points pane to add new control points to a drawing, image or map. Control points are used to georegister an image or drawing to the same geographic location and projection used by a "known good" image or drawing.

<table>
<thead>
<tr>
<th>Name</th>
<th>X / Longitude</th>
<th>Y / Latitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control1</td>
<td>-122.480997</td>
<td>37.811238</td>
</tr>
<tr>
<td>Control2</td>
<td>-122.046519</td>
<td>37.467911</td>
</tr>
</tbody>
</table>

The Control Points pane always shows a list of control points defined for the active image or drawing together with their coordinates. When we click on a different window the Control Points pane will switch context to show any control points defined for the active window. See the Georegistration topic for basic use of control points.

For more sophisticated use and management of control points we can use four controls in the Control Points pane:

- **Load Points** - Load control points from points that exist in the drawing (enabled for drawings only).
- **Save Points** - Save control points as points in the drawing (enabled for drawings only).

**Load Points**

Drawings often contain named points that can be used as control points. This saves us the extra work of adding control points for those locations.

Suppose we have a scanned image of a paper map that we would like to georegister using control points.
Suppose we also have a drawing of the same geographic region that contains points. In this example we will assume that the points represent waypoints we have acquired with a GPS device.

<table>
<thead>
<tr>
<th>Name</th>
<th>Street</th>
</tr>
</thead>
<tbody>
<tr>
<td>WP 001</td>
<td>University</td>
</tr>
<tr>
<td>WP 002</td>
<td>University</td>
</tr>
<tr>
<td>WP 003</td>
<td>University</td>
</tr>
<tr>
<td>WP 004</td>
<td>University</td>
</tr>
<tr>
<td>WP 005</td>
<td>University</td>
</tr>
<tr>
<td>WP 006</td>
<td>University</td>
</tr>
</tbody>
</table>

If we look at the table for the points we can see that each has a **Name** plus additional fields, such as the name of the street on which each waypoint is located.

We will pick out four of the waypoints to use as control points. We could use all of the points in the drawing, but for the purposes of this example we will choose only four so the screen shots do not get too cluttered with labels.

<table>
<thead>
<tr>
<th>Name</th>
<th>Street</th>
</tr>
</thead>
<tbody>
<tr>
<td>WP 043</td>
<td>Embarcadero</td>
</tr>
<tr>
<td>WP 017</td>
<td>Embarcadero</td>
</tr>
<tr>
<td>WP 006</td>
<td>University</td>
</tr>
<tr>
<td>WP 001</td>
<td>University</td>
</tr>
</tbody>
</table>
We've picked out the points named **WP 043**, **WP 017**, **WP 006**, and **WP 001**, as can be seen by looking at the table.

The Control Points pane so far is empty for this drawing.

We click on the **Load Points** button to load points from the drawing. This opens the **Load Control Points** dialog.

The **Load Control Points** dialog asks us which field to use as the name field for the control points. The list box will be loaded with all text fields available in tables associated with that drawing. We choose the field called **Name** since that is where the names of our waypoints are stored. We click **OK** to load the drawing's points as control points.

The drawing's points will be imported into the Control Points pane as control points, using the designated name field value for their **Name**.
The control points will appear in the drawing in the usual way, with labels using the imported names for each point.

To use these control points, we click on the image to be georegistered and add control points in the image at the same locations and with the same names as the control points that were added to the drawing with the **Load Points** button. The image is shown slightly desaturated and lightened so that the control point labels stand out better.

**Save Points**

After adding control points to a drawing using the Control Points pane we may wish to add them to the drawing as point objects as well.
Suppose we have added many control points to a drawing of the United Kingdom.

We can add these locations to the drawing as point objects by pressing the **Save Points** button in the Control Points pane.

The result is the creation of a new point object at every location of a control point. We can see the new points by turning off display of control points by unclicking the **Preview** button in the Control Points pane. We have formatted points in the drawing to be largish, green squares so they are more easily visible.
By opening the table associated with this drawing we can see that the **Save Points** command created new point objects and also saved the name of each control point into a text field that we specified (in this example, a field that is called "Name").

The **Save Control Points** dialog allows us to save the names of the control points into any text field in the database table associated with the drawing. The **Name** list box will display a list of all available text fields in the table that we can use for saving the name. There is also a choice available to not save the name text, in which case the control points will be created as point objects without their names being saved to any text field.

**Notes**

It does no harm to load lots of control points if there are many points in a drawing. If desired, we could have loaded dozens of control points into the drawing's Control Points pane and then specified only four control points in the image. All of the extra control points in the drawings whose names didn't match those in the image would have been ignored. Why then, did we use only four control points in the drawing in this example?

There are two main reasons: first, in real life usage it is easier to see where to place control points in an image if there are only a few control points in the drawing that need to be matched. Second, since Help file illustrations must be small in size there would have been too much visual clutter in the screen shots if we had loaded dozens of control points.

One of the illustrations above shows only four points being used in a drawing out of many original points. To create this illustration, we selected the four points desired, used a **Select Box** with **Invert** selection mode to select all the other points except these four and then deleted all the other points. In real life we would probably have copied the four selected points to a new layer instead of deleting all the unnecessary points.

The **Ignore blank values** and **Ignore duplicate values** checkboxes are disabled in the Load Control Points dialog screen shot because no point records in the example table had blank values or duplicate values.

Why is there an **Ignore duplicate values** checkbox in the Load Control Points dialog? Why would we ever want to load more than one control point with the same name? If more than one point in the drawing has the same name, we might wish to see all of the duplicates as control points. We can then easily decide which duplicates should be deleted from the control points list.

The **Native Coordinates** button was not used to toggle to regular latitude / longitude values in the screenshot of the Control Points dialog, so the **X / Latitude** and **Y / Longitude** fields were shown in the control points list using the "native" internal coordinates of the projected map.

**Very important:** Points in drawings that we might wish to use as control points will often have data fields associated with each point in which latitude or longitude values occur. When acquiring points from a GPS, for example, we will often ask the GPS console to save the latitude and longitude for each point in a data field in addition to creating a point at that location. It is very important to understand that the location of the control point is taken directly from the actual geometric location of the point within the drawing and is not taken from any fields with latitude or longitude values that happen to exist within the data table. See the **Editing Data in Tables** topic for a discussion of this issue.
For brevity, this topic and other georegistration topics use images as examples. However, the same procedures apply when georegistering drawings or surfaces.

The font used for control point labels may be changed in Tools - Options.

**Tech Tips**

When we have lists of control points measured from a paper map or taken from a GPS device it is often convenient to enter many control points by creating a table with latitude, longitude (type latitude and longitude, of course) and name (type text) fields. Each control point can then be entered into the table.

When editing fields in tables, Manifold will automatically make sense of many different ways of entering latitude and longitude data into latitude and longitude type fields. This is extremely useful when entering coordinates for control points read off from paper maps that are marked using degrees, minutes and seconds notation. See the Create a Table and Add Records example for a step by step example of how one could add control points to a table using degrees, minutes and seconds format.

Uncheck the **Autoscroll window on edit or selection operations** option in Tools - Options when working with control points. Because using the **New Control Point** tool activates the mouse as soon as it leaves the Control Points pane, if this option is not deactivated the target window will begin to autoscroll as the mouse enters it.

**See Also**

- The File - Print dialog used to print a component includes a **Control Points** option that may be used to print the control points in a component.
- See the **Control Points pane** topic for a reference guide to the pane used to assign and manage control points.
- The Georegister a Scanned Paper Map example topic shows a common georegistration task in step-by-step detail.
Manual Georegistration

Manual georegistration of a component involves moving and re-scaling the component using manual techniques by manually editing the scale and positioning of the component. This is rarely done since the automated georegistration process is far easier to use. In some rare cases, manual georegistration as set forth herein may be used to "nudge" an image slightly that has already been positioned using the automated process.

For georegistration using manual methods an image must already be arranged in a pattern equivalent to some geographic projection. The two most frequently seen such arrangements are:

- Images showing the whole world or large sections thereof with pixels arranged in Latitude / Longitude projection.
- Images showing local, overhead views that may be taken to be in Orthographic projection.

Because Manifold imports all images from non-geographic formats into Orthographic projection, manually georegistering images showing local, overhead views in North-up orientation may be done by people willing to endure the tedium involved.

Manual Georegistration using Orthographic Projection

1. Import the target image into the project.
2. Import a reference drawing that has features in common with the target image.
3. Insert a map that includes the reference drawing.
4. Click open the map and note the approximate geographic center of the drawing.
5. Drag and drop the target image into the map. Position it in a layer beneath the drawing.

Option 1

6. Click open the image in its own image window.
7. Open the Edit - Assign Projection dialog for the image. Change the Center Latitude and Center Longitude values to those measured as the center of the drawing. Press OK.
8. In the map window, the image will appear as a microscopic dot at the exact center latitude/longitude give above. Change the Local scale values for X and Y to rescale the image. Usually, the X and Y unit values should be the same and will be some fractional amount, such as one X or Y unit is .004 meters. Adjusting these values up and down in a trial and error fashion will quickly show which is the right direction to go.
9. Some images may also require a shift in Local offset in X and Y to move them sideways or up and down to match the drawing.

Option 2

6. Right click on the image's layer tab and choose Register from the context menu.
7. Using directions like those given in the Maps - Register topic, change the X unit and Y unit values to rescale the image and also change the Easting and Northing properties to move the center of the image to the center of the map. This will prove incredibly tedious and frustrating. With larger images and Preview checked it will also be annoyingly slow to boot.
There are two options to the above procedure because there are two alternative methods after step 5. The net effect of the Register dialog is to change the coordinate properties of the image. These properties may also be edited directly in the Edit - Assign Projection dialog.

See the Manually Georegister an Image topic for an example. See the Edit - Change Projection dialog topic for more on advanced projection options such as Local scale.

Notes

For fine adjustment of registration, use the layer repositioning commands.

For certain expert uses it makes sense to use manual georegistration. For almost all other purposes, the automated georegistration process employing control points is far easier to use.

For brevity, this topic and other georegistration topics use images as examples. However, the same procedures apply when georegistering drawings or surfaces.

Projections Readings

The three topics in the Projections Readings section are based on the writings of John Parr Snyder as published by the U.S. Geological Survey. They have been liberally supplemented with writings contributed by the manifold.net team. The intent is to provide access to a classical view of projections and their usage as expressed by Snyder.

Much of this material is taken word-for-word from Map Projections Used by the U.S. Geological Survey, Geological Survey Bulletin 1532, Second Edition, John P. Snyder, 1982. However, it has been liberally revised by the current author so any errors that have been introduced should not be attributed to Snyder.

General Projections Concepts
The Earth as an Ellipsoid
Guide to Selecting Map Projections
General Projections Concepts
For almost 500 years, it has been conclusively established that the Earth is essentially a sphere, although there were a number of intellectuals nearly 2,000 years earlier who were convinced of this. Even to the scholars who considered the Earth flat, the skies appeared hemispherical, however. It was established at an early date that attempts to prepare a flat map of a surface curving in all directions leads to distortion of one form or another.

A map projection is a device for reproducing all or part of a round body on a flat sheet. Since this cannot be done without distortion, the cartographer must choose the characteristic that is to be shown accurately at the expense of others, or a compromise of several characteristics. There is literally an infinite number of ways in which this can be done, and several hundred projections have been published, most of which are rarely used novelties. Most projections may be infinitely varied by choosing different points on the Earth as the center or as a starting point. Manifold, for example, allows the setting of the projection’s center point for most projections.

It cannot be said that there is one "best" projection for mapping. It is even risky to claim that one has found the "best" projection for a given application, unless the parameters chosen are artificially constricting. Even a carefully constructed globe is not the best map for most applications because its scale is by necessity too small, a straightedge cannot be satisfactorily used on it for measurement of distance, and it is awkward to use in general.

The characteristics normally considered in choosing a map projection are as follows:

Area - Many map projections are designed to be equal-area, so that a coin, for example, on one part of the map covers exactly the same area of the actual Earth as the same coin on any other part of the map. Shapes, angles, and scale must be distorted on most parts of such a map, but there are usually some parts of an equal-area map which are designed to retain these characteristics correctly, or very nearly so. Less common terms used for equal-area projections are equivalent, homolographic, authalic, and equiareal.

Shape - Many of the most common and most important projections are conformal or orthomorphic, in that normally the shape of every small feature of the map is shown correctly. On a conformal map of the entire Earth there are usually one or more "singular" points at which shape is still distorted. A large landmass must still be shown distorted in shape, even though its small features are shaped correctly. An important result of conformality is that relative angles at each point are correct, and the local scale in every direction around any one point is constant. Consequently, meridians intersect parallels at right (90 degree) angles on a conformal projection, just as they do on the Earth. Areas are generally enlarged or reduced throughout the map, but they are relatively correct along certain lines, depending on the projection. Nearly all large-scale maps of the Geological Survey and other mapping agencies throughout the world are now prepared on a conformal projection.

Scale - No map projection shows scale correctly throughout the map, but there are usually one or more lines on the map along which the scale remains true. By choosing the locations of these lines properly, the scale errors elsewhere may be minimized, although some errors may still be large, depending on the size of the area being mapped and the projection. Some projections show true scale between one or two points and every other point on the map, or along every meridian. They are called equidistant projections.

Direction - While conformal maps give the relative local directions correctly at any given point, there is one frequently used group of map projections, called azimuthal or zenithal, on which the directions or azimuths of all points on the map are shown correctly with respect to the center. One of these projections is also equal-area, another is conformal, and another is equidistant. There are also projections on which directions from two points are correct, or on which directions from all points to one or two selected points are correct,
but these are rarely used.

**Special Characteristics** - Several map projections provide special characteristics that no other projection provides. On the Mercator projection, all rhumb lines, or lines of constant direction, are shown as straight lines. On the Gnomonic projection, all great circle paths - the shortest routes between points on a sphere - are shown as straight lines. On the Stereographic, all small circles, as well as great circles, are shown as circles on the map. Some newer projections are specially designed for satellite mapping. Less useful but mathematically intriguing projections have been designed to fit the sphere conformally into a square, an ellipse, a triangle, or some other geometric figure.

**Method of Construction** - In the days before ready access to computers, ease of construction was of greater importance. Some projections have become popular simply because they are easy to compute. With the advent of computers, very complicated formulas can be handled as routinely as simple projections in the past.

While the above features should ordinarily be considered in choosing a map projection, they are not so obvious in recognizing a projection. In fact, if the region shown on a map is no much larger than the United States, for example, even a trained eye cannot often distinguish whether the map is equal-area or conformal. It is necessary to make measurements to detect small differences in spacing or location of meridians and parallels, or to make other tests. The type of construction of the map projection is more easily recognized with experience, if the projection falls into one of the common categories.

**Categories of Projections**

A **developable** surface is one that can be transformed to a plane without distortion. There are three types of developable surfaces onto which most of the map projections used by USGS and other agencies are at least partially geometrically projected. They are the **cylinder**, the **cone**, and the **plane**. Actually all three are variations of the cone. A cylinder is a limiting form of a cone with an increasingly sharp point or apex (i.e., drawn out to infinity). As the cone becomes flatter, its limit is a plane.

If a cylinder is wrapped around the globe representing the Earth, so that its surface touches the Equator throughout its circumference, the meridians of longitude may be projected onto the cylinder as equidistant straight lines perpendicular to the Equator, and the parallels of latitude marked as lines parallels to the Equator, around the circumference of the cylinder and mathematically spaced for certain characteristics. When the cylinder is cut along some meridian and unrolled, a cylindrical projection with straight meridians and straight parallels results. The Mercator projection is the best-known example.

Regular Cylindrical Projection

If a cone is placed over the globe, with its peak or apex along the polar axis of the Earth and with the surface of the cone touching the globe along some particular parallel of
latitude, a conic (or conical) projection can be produced. This time the meridians are projected onto the cone as equidistant straight lines radiating from the apex, and the parallels are marked as lines around the circumference of the cone in planes perpendicular to the Earth's axis, spaced for the desired characteristics.

![Regular Conic Projection](image)

When the cone is cut along a meridian, unrolled, and laid flat, the meridians remain straight radiating lines, but the parallels are now circular arcs centered on the apex. The angles between meridians are shown smaller than the true angles.

A plane tangent to one of the Earth's poles is the basis for polar azimuthal projections. In this case, the group of projections is named for the function, not the plane, since all common tangent-plane projections of the sphere are azimuthal. The meridians are projected as straight lines radiating from a point, but they are spaced at their true angles instead of the smaller angles of the conic projections. The parallels of latitude are complete circles, centered on the pole.

![Polar Azimuthal Projection](image)

On some important azimuthal projections, such as the Stereographic (for the sphere) the parallels are geometrically projected from a common point of perspective; on others, such as the Azimuthal Equidistant, they are non-perspective.

The concepts outlined above may be modified in two ways, which still provide cylindrical, conic, or azimuthal projections (although the azimuthals retain this property precisely only for the sphere, not for ellipsoidal Earth models):

- The cylinder or cone may be secant to or cut the globe at two parallels instead of being tangent to just one. This conceptually provides two standard parallels (as settable in some Manifold projections); but for most conic projections this construction is not geometrically correct. The plane may likewise cut through the globe at any parallel instead of touching a pole. Those Manifold projections which allow secant projection surfaces will allow the setting of additional standard parallels beyond what is required for the simple tangent form of the projection.
- The axis of the cylinder or cone can have a direction different from that of the Earth's axis, while the plane may be tangent to a point other than a pole. This type of modification leads to important oblique, transverse and Equatorial projections, in which most meridians and parallels are no longer straight lines or arcs of circles. What were standard parallels in the normal orientation now become standard lines not following parallels of latitude.
Some other projections in common use resemble one or another of these categories only in some respects. The Sinusoidal projection is called pseudocylindrical because its latitude lines are parallel and straight, but its meridians are curved. The Polyconic projection is projected onto cones tangent to each parallel of latitude, so the meridians are curved, not straight. Still others are more remotely related to cylindrical, conic, or azimuthal projections, if at all.

**Projection Names**

Manifold System includes a vast array of different projections that are named mostly in accordance as described in US Geological Survey bulletins which tend to follow international cartographic practise for the names of the most common projections.

Many of the "standard" projections allow the use of various projection parameters as described above. Some countries have standardized on the use of a particular projection for mapping their countries that has acquired a local name when used with the locally-preferred set of parameters. Manifold includes these "projections" as choices for many of the more well known national projections.

Datum names used within Manifold originate with the U.S. National Imagery and Mapping Agency the official keeper of such data for the U.S. government.
The Earth as an Ellipsoid

For many maps, including nearly all maps in commercial atlases, it may be assumed that the Earth is a sphere. Actually, it is more nearly a slightly flattened sphere - an oblate ellipsoid of revolution, also called an oblate spheroid. This is an ellipse rotated about its shorter axis. The flattening of the ellipse for the Earth is only about one part in three hundred; but it is sufficient to become a necessary part of calculations in plotting accurate maps at a scale of 1:100,000 or larger, and is significant even for 1:5,000,000-scale maps of the United States, affecting plotted shapes by up to 2/3 percent. On small-scale maps, including single-sheet world maps, the oblateness is negligible.

The bad news is that the Earth is not an exact ellipsoid. In fact, because the Earth is such a "lumpy" ellipsoid no single smooth ellipsoid will provide a perfect reference surface for the entire Earth. The practical solution to this is to measure the Earth's shape in different areas and to then create different reference ellipsoids used for mapping different regions on Earth.

For example, the ellipsoid shown in yellow above is a fair match to the Earth's surface (shown in blue) in some areas but not in others. In some areas the Earth's surface protrudes above the even ellipsoid shape and in other areas the Earth's surface is lower than the ellipsoid's surface. We can use the yellow ellipsoid for precision mapping in areas where the Earth's surface is a close match.
We can use a different ellipsoid (shown in magenta) to map other areas where the magenta ellipsoid is a better match to the Earth's surface. We can specify many different standard ellipsoids to map different areas of the Earth.

**Offset Ellipsoids**

One more refinement lets us use a standard collection of ellipsoids without having to create hundreds of different reference ellipsoids to fit all the different lumpy regions of the Earth. The refinement is to use the same ellipsoid in different areas, but to offset the ellipsoid slightly to make it a better match. For example, in the illustration above we could use the magenta ellipsoid as is for mapping northern regions and we could also use it in southern regions if we moved it up slightly.

To see how this works, consider a translucent Earth in blue color with the center of the Earth marked by the intersection of three green axes.
For any ellipsoid we choose to use we can mark the center of the ellipsoid as well. The ellipsoid is shown in dotted outline with yellow axes marking the center.

To achieve a better fit between a given ellipsoid and a particular region of the Earth we can offset the standard ellipsoid from the center of the Earth.
If we illustrate the situation with a wireframe ellipsoid and solid Earth we can see the ellipsoid is above the Earth's surface in some areas and below it in others. In those regions where the ellipsoid closely follows the surface of the Earth we can use it for high-accuracy mapping.

To get it to fit better in other areas we can move the ellipsoid relative to the Earth.

A datum is a reference ellipsoid together with an offset from the center of the Earth. By specifying different offsets, we can use the same standard ellipsoids in many different regions of the Earth. Different countries will often use the same ellipsoid but with different offsets for standard government maps in those countries. In Manifold, each such unique combination is listed as a different datum in the Datum choice box in Manifold projection dialogs. There are literally hundreds of choices.

There are over a dozen principal ellipsoids that are frequently used by one or more countries. The Datum choice box in Manifold projections dialogs will call up a table of hundreds of datums that each use some standard ellipsoid. The different dimensions for various standard ellipsoids result from varying accuracy in geodetic measurements (the
measurements of locations on the Earth). Differences also arise because the curvature of the Earth’s surface is not uniform due to irregularities in the gravity field. Therefore the ellipsoid one calculates to be a best fit for the surface of the Earth depends not only on how the measurements are made but also where they are made.

Until recently, ellipsoids were only fitted to the Earth’s shape over a particular country or continent so that in effect every datum used in various countries employed an offset ellipsoid as illustrated above. The discrepancy between centers is usually a few hundred meters at most. In more recent years satellite-determined coordinate systems, such as the WGS series, have resulted in geocentric ellipsoids. The center of a geocentric ellipsoid is the same as the center of the Earth. Satellite-computed geocentric ellipsoids represent the entire Earth more accurately on an overall, average basis than ellipsoids determined from ground measurements but they do not generally give the "best fit" for a particular region.

The hundreds of choices for the datum in Manifold projection dialogs arise because many different countries over the years have used hundreds of different combinations of "standard" ellipsoids with different offsets. When creating new maps, pick the datum that is the one most used by other maps with which you will work. For general-purpose mapping, the safest choice is the Manifold default, the WGS 84 World Geodetic datum used for almost all GPS work.

**Historical Notes**

A frequently-used official Earth ellipsoid was defined in 1924, when the International Union of Geodesy and Geophysics (IUGG) adopted a flattening of exactly 1 part in 297 and a semi-major axis (or Equatorial radius) of exactly 6,378,388 m. The radius of the Earth along the polar axis is then 1/297 less than 6,378,388 or approximately 6,356,911.9 m. This is called the International ellipsoid and is based on John Fillmore Hayford’s calculations in 1909 from U.S. Coast and Geodetic Survey measurements made entirely within the United States. Although this datum is most accurate in the US it is amusing to note that this ellipsoid was adopted for international use and not adopted for use in North America. The datums still used in many countries employ the International ellipsoid together with a particular offset that best aligns the International ellipsoid with the Earth’s surface in the region of that particular country.

**The North American Datum of 1927 / NAD 27**

The first official geodetic datum in the United States was the New England Datum, adopted in 1879. It was based on surveys in the Eastern and Northeastern states and referenced to the Clarke ellipsoid of 1866, with triangulation station Principio, in Maryland, as the origin. The first transcontinental arc of triangulation was completed in 1899, connecting independent surveys along the Pacific Coast of the U.S. In the intervening years, other surveys were extended to the Gulf of Mexico. The New England Datum was thus extended to the south and west without major readjustment of the surveys in the east. In 1901, this expanded network was officially designated the United States Standard Datum, and triangulation station Meades Ranch, in Kansas, was the origin. In 1913, after the geodetic organizations of Canada and Mexico formally agreed to base their triangulation networks on the United States network, the datum was renamed the North American Datum.

By the mid-1920’s, the problems of adjusting new surveys to fit into the existing network were acute. Therefore, during the 5-year period 1927-1932 all available primary data were adjusted into a system now known as the North American 1927 Datum. The coordinates of station Meades Ranch were not changed but the revised coordinates of the network comprised the North American 1927 Datum.
Recent Ellipsoids

The ellipsoid adopted for use in North America is the result of the 1866 evaluation by the British geodesist Alexander Ross Clarke using measurements made by others of meridian arcs in western Europe, Russia, India, South Africa, and Peru. This resulted in an adopted Equatorial radius of 6,378,206.4 m and a polar radius of 6,356,583.8 m, or an approximate flattening of 1/294.9787. Since Clarke is also known for an 1880 revision used in Africa, the Clarke 1866 ellipsoid is named with the year. Once again it is amusing to note that the ellipsoid used for the North American Datum is based on data compiled outside of North America and thus is a less accurate choice than the International ellipsoid.

Satellite tracking data have provided geodesists with new measurements to define the best Earth-fitting ellipsoid and for relating existing coordinate systems to the Earth’s center of mass. The Defense Mapping Agency’s efforts produce the World Geodetic System 1966 (WGS 66) followed by more recent evaluations (WGS 72, WGS 84).

The North American 1927 Datum has been replaced with a new datum, the North American Datum 1983 (NAD 83) that is Earth-centered based on satellite tracking data, using the Geodetic Reference System 1980 (GRS 80) ellipsoid, an ellipsoid very similar to that for the WGS 72.

Eccentricity

In most map projection formulas, some form of the eccentricity $e$ is used rather than the flattening $f$ ratio. Manifold can accept the specification of a custom ellipsoid using the Major axis together with either the Minor axis or eccentricity.

Other Planets and Satellites

For the mapping of other planets and natural satellites, only Mars and Earth among the inner planets is treated as an ellipsoid. The Moon, Mercury, Venus, and the satellites of Jupiter and Saturn are taken as spheres. Manifold provides parameters for all the planets except Pluto.

Numbers for radius (Major axis and minor axis are the same for a sphere) for many planets and satellites may be found from a variety of Internet sites. Recent planetary missions have greatly improved accuracy.

Notes

The illustrations for this topic were created in Manifold 3D View Studio. They are greatly exaggerated to illustrate the concepts. The actual differences between ellipsoids and the Earth’s surface are rarely more than a few hundred meters.

It is difficult for humans to realize just how smooth the surface of the Earth really is on a large scale. Although we are impressed by the dramatic rise of mountains on a human scale, if the Earth were reduced to the size of a billiard ball the Earth would be much smoother than the billiard ball. The difference in height between the median surface and even the Himalayan Mountains is much less than the variations in smoothness of billiard balls.

[About half of this topic is taken word-for-word from Map Projections Used by the U.S. Geological Survey, Geological Survey Bulletin 1532, Second Edition, John P. Snyder, 1982. However, it has been liberally revised by the current author so any errors that have been introduced should not be attributed to Snyder.]
Guide to Selecting Map Projections
(This topic is based almost verbatim on the text in An Album of Map Projections, U.S. Geological Survey Professional Paper 1453, by John P. Snyder. It is included so Manifold users can benefit from the words of the master himself.)

The advent of computer-assisted cartography has now made it much easier to prepare maps. A map can now be centered anywhere on the globe, can be drawn according to any one of many projection formulas, and can use cartographic data files having a level of detail appropriate to the scale of the map.

Properties of Map Projections

An equal-area map projection correctly represents areas of the sphere on the map. If a coin is placed on any area of such a map, it will cover as much of the area of the surface of the sphere as it would if it were placed elsewhere on the map. When this type of projection is used for small-scale maps showing larger regions, the distortion of angles and shapes increases as the distance of an area from the projection origin increases.

An equidistant map projection is possible only in a limited sense. That is, distances can be shown at the nominal map scale along a line from only one or two points to any other point on the map. The focal points usually are at the map center or some central location. The term is also often used to describe maps on which the scale is shown correctly along all meridians.

An azimuthal map likewise is limited in the sense that it can correctly show directions or angles to all other points on the map only with respect to one (or rarely two) central point(s).

A conformal map is technically defined as a map on which all angles at infinitely small locations are correctly depicted. A conformal projection increasingly distorts areas away from the map’s center point or lines of true scale and it increasingly distorts shapes as the region becomes larger but distorts the shapes of moderately small areas only slightly.

Consistent with these definitions, maps simultaneously exhibiting several of these properties can be devised:

<table>
<thead>
<tr>
<th></th>
<th>Conformal</th>
<th>Equal Area</th>
<th>Equidistant</th>
<th>Azimuthal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conformal</td>
<td>-</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Equal Area</td>
<td>No</td>
<td>-</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Equidistant</td>
<td>No</td>
<td>No</td>
<td>-</td>
<td>Yes</td>
</tr>
<tr>
<td>Azimuthal</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>-</td>
</tr>
</tbody>
</table>

A map projection may have none of these general properties and still be satisfactory. For example, the Robinson projection is well suited for thematic presentations involving the entire Earth, but it has none of the above properties.

A map projection possessing one of these properties may nevertheless be a poor choice. As an example, the Mercator projection continues to be used inappropriately for world-wide thematic data. The Mercator map projection is conformal and has a valid use in navigation but very seriously distorts areas near the poles, which it cannot even show. It should not be used (although it frequently is) for depicting general information or any area-related subjects.
Some Nomenclature

From the perspective of design as well as distortion reduction, a projection may be selected because of the characteristic curves formed by the meridians and parallels. Using nomenclature suggested by L.P. Lee, we use the following terms to describe projections:

**Cylindric:** Projections in which the meridians are represented by a system of equidistant parallel straight lines, and the parallels by a system of parallel straight lines at right angles to the meridians.

**Pseudocylindric:** Projections in which the parallels are represented by a system of parallel straight lines and the meridians by concurrent curves.

**Conic:** Projections in which the meridians are represented by concentric circular arcs and the meridians by concurrent curves.

**Pseudoconic:** Projections in which the parallels are represented by concentric circular arcs, and the meridians by concurrent curves.

**Polyconic:** Projections in which the parallels are represented by a system of nonconcentric circular arcs with their centers lying on the straight line representing the central meridian.

**Azimuthal:** Projections in which the meridians are represented by a system of concurrent straight lines inclined to each other at their true difference of longitude, and the parallels by a system of concentric circles with their common center at the point of concurrency of the meridians.

Map Projection Illustrations

**Note:** In a few cases, projection is geometric, but in most cases the projection is mathematical to achieve certain features.

Regular Cylindrical - Specifying latitude and longitude origin centers the projection about a given location. Various cylindrical projections allow the specification of standard parallels that are used to govern the mathematical mapping from sphere to cylinder.
Transverse Cylindrical - Specifying latitude and longitude origin (or in some cases, the standard meridian) centers the projection about a given location. Various transverse cylindrical projections allow the specification of standard parallels that are used to govern the mathematical mapping from sphere to cylinder.

Oblique Cylindrical - Specifying latitude and longitude origin (or in some cases, the standard meridian) centers the projection about a given location.

Regular Conic - Illustrated with the latitude and longitude origin at the pole. Specifying a lat/lon origin other than the pole results in an oblique projection. Standard parallels, if specified, define the mathematical mapping from ellipsoid to the cone.

Polar Azimuthal (plane) - The projection’s latitude and longitude origin is at the pole.

Oblique Azimuthal (plane) - The projection is centered at a latitude and longitude origin other than the pole.

Philosophy of Map Projection Selection
Three traditional rules for choosing a map projection were at one time recommended as follows:

- For low-latitude areas: cylindrical.
- For middle-latitude areas: conical.
- For Polar Regions: azimuthal.

Inherent in these guidelines was the idea that it would be difficult to re-center a map so that the area of main interest was near the area of the map that has the least areal or angular distortion. On the other hand, with Manifold it is easy to recenter the projection to any location.

As a result, a projection no longer needs to be rejected merely because previous uses were traditionally centered inappropriately for the desired application.

In fact, the mathematical form of many projections (as implemented in Manifold) permit the user to alter the form of the map to reduce the distortions within a certain area. Most commonly, such alteration is accomplished by establishing standard lines along which distortion is absent; often, these lines are parallels of latitude. The Albers Equal-Area Conic projection, for example, will often be used with two standard parallels to “customize” the projection to a particular region. However, most properties of the map projection are affected when a standard line is changed. One should therefore keep track of what parameters were used in a particular projection.

Another way of altering the relationships on a map is by using different aspects, which involves moving the center of the projection from the normal position at a pole or along the Equator to some other position. As with the case of custom standard lines, one should keep track of any re-centering done with a given projection.

When a map requires a general property, the choice of a projection becomes limited. For example, because conformal projections correctly show angles at every location, they are advisable for maps displaying the flow of oceanic or atmospheric currents. The risk of using a conformal projection for a world-wide map is that the distortion of areas greatly enlarges the outer boundaries, and a phenomenon may seem to take on an importance that the mapmaker did not intend. Equal-area maps should be considered for displaying area-related subjects or themes, such as crop-growing regions.

Once the purpose of a map has been decided, the geographical area to be included on the map must be determined. This may be a region, or the entire world. The larger the area covered, the greater is the Earth’s curvature involved in the map. If the map area is a region, then its shape, size and location are important determinants in making decisions concerning projections.

Bearing in mind these determinants, mapmakers can apply traditional rules of choice, such as those mentioned above, or they can study the patterns of distortion associated with particular projections. For example, azimuthal projections have a circular pattern for lines of constant distortion characteristics, centered on the map origin or projection center. Thus, if an area is approximately circular and if its center is made the origin of the projection, it is possible to create a map that minimizes distortion for that map area. Ideally, the general shape of a geographic region should be matched with the distortion pattern of a specific projection.

An appropriate map projection can be selected on the basis of these principles and classifications. Although there may be no absolutely correct choice, it is clearly possible to make a bad judgement.
Addendum: Suggestions from the Manifold Team

Mapping experts will choose from a wide array of projections to meet specific objectives as discussed above. For general use, most people will usually develop a favorite projection for a particular situation and then use that projection over and over. For most situations either Orthographic or Lambert Conformal Conic are good choices.

The Manifold team recommends the following:

<table>
<thead>
<tr>
<th>Region to be Mapped</th>
<th>Suggested Projections</th>
</tr>
</thead>
<tbody>
<tr>
<td>Whole Earth</td>
<td>Robinson (pseudocylindrical) or Miller Cylindrical. Robinson seems to be fashionable for thematic maps. Any of the pseudocylindrical projections will be fine if you like their appearance better.</td>
</tr>
<tr>
<td>Hemispheres</td>
<td>Orthographic (azimuthal) for a &quot;view from space&quot; look, and Lambert Azimuthal Equal Area for thematic maps where the relative size of countries near the edge of the projection is to be preserved.</td>
</tr>
<tr>
<td>Continents</td>
<td>Use Lambert Conformal Conic for North America and Eurasia. Use Lambert Azimuthal Equal Area or Orthographic for South America and Africa. Use Orthographic for Australia, and Antarctica.</td>
</tr>
<tr>
<td>E-W Countries or Regions</td>
<td>Use Lambert Conformal Conic for US, Canada, Russia, and China. Use either Lambert Conformal Conic or Orthographic for Europe. Use Orthographic or Lambert Azimuthal Equal Area otherwise.</td>
</tr>
<tr>
<td>Polar Regions</td>
<td>Orthographic or Lambert Azimuthal Equal Area.</td>
</tr>
<tr>
<td>Oceans</td>
<td>Orthographic or Lambert Azimuthal Equal Area.</td>
</tr>
<tr>
<td>Smaller Countries or Regions</td>
<td>Orthographic.</td>
</tr>
<tr>
<td>N-S Countries, Oblique Regions</td>
<td>Long, thin countries aligned North-South such as Chile are one of the few times we would use Transverse Mercator. Oblique regions like the Alaska panhandle are mercifully rare: Use the Oblique Mercator in such cases.</td>
</tr>
</tbody>
</table>

Other than personal taste in visual appearance there are only two reasons not to use one of the above projections:

- A specific projection is required to match other data available or to exchange data with other users.
- The technical characteristics of a particular projection such as preservation of scale or relative area are required for creating a paper map or other presentation.

Manifold provides a very wide array of projections in addition to the standard projections.
mentioned above.

Frequently Asked Question

If Orthographic or Lambert Conformal Conic work well for almost all mapping that does not involve the entire world, why are systems like Universal Transverse Mercator (UTM) so frequently encountered? These systems came into use in a day when any mapping or projection computation had to be painfully prepared by hand. They were developed to solve problems within the technological limits of their day. They live on in modern times as living fossils simply because so many maps have been prepared using them that their usage has gained momentum. In some cases, the use of projections like UTM is required by law.

Modern desktop computers make it possible to compute a precise, perfectly centered projection for any location on Earth so in modern times we can use whatever projection works best for all the factors we wish to consider. We can completely ignore the cost of projection since there isn't any cost. We can therefore choose projections that minimize distortion such as the Lambert Conformal Conic or provide a very natural look like the Orthographic, and compute each projection to fit perfectly the view desired.

Manifold Projections

Manifold Projections

Because there is no perfect way of representing a three-dimensional Earth on two-dimensional paper maps or computer screens, many different projections have been invented that trade off various advantages and disadvantages. It is important to remember that every projection has some disadvantage and inaccuracy to it. When using projections, the key to choosing the right projection is to choose so that the disadvantages of that particular projection are irrelevant to the proposed usage of our map while the advantages serve our purposes.

Manifold includes numerous standard types of projections that with optional parameters may be used to specify hundreds of standard projections. For example, choosing a UTM zone in the Universal Transverse Mercator (UTM) projection series will specify any one of 60 specific projections in the UTM series.

Within most projection types, optional parameters may be used to create an effectively infinite number of different projections. Even standard projection types can exist in thousands of different variations when optional parameters such as datum are changed.

The individual projection topics introduce each main type of projection with a summary of the main characteristics, usage and origin of the projection. The projection descriptions are taken (almost verbatim, in some cases) from the writings of John Parr Snyder (1926 - 1997) of the USGS, most notably "An Album of Map Projections" by Snyder and Voxland (USGS Professional Paper 1453). Additional descriptions have been taken from the wonderful summary poster by Tau Rho Alpha and John P. Snyder "The Properties and Uses of Selected Map Projections", 1982, USGS Map I-1402.

All projection formulas used within Manifold have been programmed by the manifold.net team using the full ellipsoidal formulae that define the projection, wherever applicable. The main source of projection formulae is J.P. Snyder's definitive work, "Map Projections - A Working Manual" (USGS Professional Paper 1395).

Manifold forces the results of a coordinate system (projection) transformation to stay finite. This allows working with maps containing layers with locations which can not be represented in the coordinate system of the map. For example we can work with a
Mercator map that contains a drawing in latitude / longitude of the Antarctic region with objects touching a latitude of -90, which otherwise would have to be projected to infinity.

**Automatic use of Custom Datum Transformations**

Manifold uses high accuracy coordinate transformation mathematics when re-projecting data from one coordinate system to another. In certain parts of the world, custom transformation formulae are routinely used (and, at times mandated by law for certain uses) to convert datums during re-projection. If the **Use custom datum transformations** option is checked in the Tools - Options - Miscellaneous pane (checked by default) Manifold will use such custom datum transformations when available.

Currently, **NADCON** formulae are used to convert between **NAD27** and **NAD83** in North America and **NTv2** formulae are used to convert Canadian, Australian and New Zealand datums supported by **NTv2**.

If a converted location is outside of the conversion domain supported by such custom methods, no datum conversion is done and Manifold will display a message box to that effect after the attempted re-projection. If desired, such a re-projection can be undone, the data set edited to fall entirely within the region supported by the custom method and re-projection attempted once more.

**Note:** the message box mentioned above appears only when a component is re-projected using **Edit-Projection**. It does not appear when a component is re-projected on-the-fly for display as part of a map view.

In general, if we attempt to make a datum conversion between any of the datum pairs shown below and the datum is not converted, that tells us part of the data set we are attempting to convert falls outside the service area of the custom transformation formulae.

**Datum Pairs supported by Custom Transformations**

| North American 1927 (mean for CONUS) | **< NADCON >** | North American 1983 (mean for CONUS) |
| North American 1927 (mean for Canada) | **< NTv2 >** | North American 1983 (mean for CONUS) |
| Australian Geodetic 1966 (Australia, Tasmania) | **< NTv2 >** | Australian Geocentric 1994 (GDA94) |
| New Zealand Geodetic 1949 (New Zealand) | **< NTv2 >** | New Zealand Geodetic 2000 |

Custom datum transformations will be invoked **only** when converting between the above pairs in either direction. For example, they will be used when converting NAD27 to NAD83, or when converting from NAD83 to NAD27. Pairs shown above are listed using their full names as they appear in the datum box in Manifold projection dialogs, not using the abbreviated form commonly used. For example, **North American 1983 (mean for CONUS)** is referred to as **NAD83** in abbreviated form.

The **names** of the above datums are significant, not the parameters used. For example, if we convert **GRS80** (the parameters of which are identical to NAD83 so it is the same datum) to **NAD27** (mean for Canada), Manifold will **not** use the custom NADCON conversion formulae. This is done on purpose, so we can use NAD83 as the name of the datum for conversion-sensitive North American data (so that the custom transformation will be used).
and GRS80 as the name of the datum for all other data where we would like to use the
general purpose routines.

Custom Datum Pairs for NTv2

Additional datum pairs may be added to those supported by Manifold for use in NTv2
transformations through customization. See the Custom Datum Grids for NTv2 topic for
details.

Organization

There are so many projections available within Manifold that they have been organized into
several classes in the Edit - Assign Projection tree diagram:

- **Gauss Kruger**: A Universal Transverse Mercator projection
defined in zones and used in Germany, South
America, the former USSR and various
Eastern European nations. The European
alternative to UTM.

- **National Grids**: Presets for various national grid systems,
including the US State Plane Coordinate
System using either NAD27 or NAD83.

- **Standard**: Selection of standard Azimuthal, Conic,
Cylindrical, Pseudocylindrical, Satellite
View and Latitude / Longitude projections.

- **Universal Transverse Mercator**: The Northern Hemisphere projections for the
infamous UTM system consisting of 120
zones (60 different zones with North and
South variants of each). Originally developed
for military use and now widely misused in
civil mapping.

- **Universal Transverse Mercator (South)**: The Southern Hemisphere projections for
UTM. These are mainly distinguished by
each having a Northing parameter of 10
million so that no coordinates need involve
negative numbers.

Datums

There are hundreds of datums available within Manifold. They are listed alphabetically by
their official names as classed by the United States government agency (NIMA) current in
charge of such things. Each datum incorporates a specific Earth ellipsoid together with a
specific datum offset.

For general mapping use the **World Geodetic 1984 (WGS84)** datum. It uses an "Earth
centered" ellipsoid that provides a good match when the entire Earth is considered and is
the standard for GPS work.

Clip Coordinates Option

Some projections, like the Orthographic have a **clip coordinates** check box that clips
invisible parts of objects, such as countries located on the other side of the Earth from the
point of view of the projection. Clipping is a destructive process in that objects being
clipped are permanently removed. If the component might at a future date be used in a
different projection, make a copy before projecting using **clip coordinates**.
**Customization**

Experts may specify custom units of measure, ellipsoids, datums and even customized coordinate systems (projections) by specifying customized projection presets. See the Customization topic.

**Note**

Take a moment to purchase "An Album of Map Projections" by John P. Snyder and Philip M. Voxland (USGS Professional Paper 1453) from USGS. At $20 it's a steal and worth every penny for the numerous, clear illustrations of various projections. This and the other USGS publications noted above should be in every serious Manifold user's library.

**Gauss Kruger**

A system using Transverse Mercator projections to map the world into numerous standard zones that are six degrees wide. The standard Manifold Gauss-Kruger projection is also known as the Pulkovo 1942 Gauss-Kruger projection. A similar Gauss-Kruger projection is implemented in the **National Grids** group of Manifold projections under Germany as the **DHDN** Gauss-Kruger projection presets.

Americans can best understand the Gauss Kruger system by thinking of it as a version of the **Universal Transverse Mercator (UTM)** system with zones defined for European and Asian coverage. (Europeans, of course, can think of the UTM system as a version of the Gauss Kruger system that has been implemented by Americans.)

See the Universal Transverse Mercator (UTM) topic for information on how systems such as UTM and Gauss Kruger operate.

**Comments**

Like UTM or the State Plane Coordinate System, the Gauss Kruger system of projections is a living fossil. It was created mostly as a reflection of the technological limitations of an earlier era. When used with skill as originally intended it still functions well in expert hands. The problem with the Gauss Kruger system (as with UTM or the State Plane) is that it is constantly misused in civil applications by inexpert users who do not realize the limitations built into the system.

**Graphics**

**Graphics Projections**
Manifold includes several coordinate system (projection) presets that are useful when working with images in graphics editing that does not involve any geographic context. The graphics projection presets configure the default Orthographic projection with **Local scale** and units so pixel sizes correspond to typical sizes used at various DPI (dots per inch) settings commonly used for monitor displays or scanning. The graphics projections are normally assigned using the **Edit - Assign Projection** dialog to images imported from non-geographic formats.
There are four resolutions (72 DPI, 96 DPI, 150 DPI and 300 DPI) and two variations for measurement units (inches and millimeters). Be sure to uncheck the **Preserve local values** box when switching to these presets.

**Assigning a graphics projection to an image:**

1. Import the image.
2. Open the image.
3. Launch the *Edit - Assign Projection* dialog.
4. In the *Assign Projection* dialog, uncheck the **Preserve local values** box.
5. Choose the desired DPI and units and press **OK**.

The main value of the graphics projections is that the size shown by the image when using View - Zoom To - Native will be the "true" size of the image given the DPI intended. In addition, the size reported by selection tools and the Tracker will be accurate for the intended size of the image.

**Tech Tip**

*Advanced users desiring resolutions other than those provided may create their own presets using settings similar to those provided. See the Custom Coordinate System (Projection) Presets topic.*

**National Grids**

*National Grids*

National grid systems are set up by countries to provide a standard scheme of map projections for their country. Such systems might use one standard projection type (such as UTM) with specific parameters for standard zones within the country.

Other national grid systems will use different projections for different parts of the country. The US State Plane Coordinate System, for example, uses UTM as well as Lambert Conformal Conic in different State Plane zones.

Manifold includes standardized projections for a vast number of national grids. All have been implemented using local standards for such grids. The detailed description of such grids and how to use them is beyond the scope of this documentation. When working with a particular country's national grid, consult the references published by the national cartographic authority.

Following is a partial list of national grid projection systems in Manifold:
<table>
<thead>
<tr>
<th>Country</th>
<th>Projections</th>
</tr>
</thead>
<tbody>
<tr>
<td>Australia</td>
<td>AMG66, AMG84, MGA94 and NSW ISG 56:3.</td>
</tr>
<tr>
<td>Austria</td>
<td>Austria LCC, M28, M31 and M34.</td>
</tr>
<tr>
<td>Bahrain Grid</td>
<td></td>
</tr>
<tr>
<td>Barbados Grid</td>
<td></td>
</tr>
<tr>
<td>Belgium</td>
<td>Hooijberg and National Geographic Institute projections.</td>
</tr>
<tr>
<td>British National Grid</td>
<td></td>
</tr>
<tr>
<td>Canada (NAD 27)</td>
<td>MTM Zones 1 through 6 using NAD 27.</td>
</tr>
<tr>
<td>Canada (NAD 83)</td>
<td>MTM Zones 1 through 6 using NAD 83.</td>
</tr>
<tr>
<td>Dutch Grid</td>
<td>Finland Uniform Coordinate System and Finland Zones 1 through 4</td>
</tr>
<tr>
<td>(Rijksdriehoekstelsel Grid)</td>
<td></td>
</tr>
<tr>
<td>Finnish Grid</td>
<td></td>
</tr>
<tr>
<td>France</td>
<td>Entire France, Zone I (Nord), Zone II (Midi), Zone III (Sud) and Zone IV (Corse).</td>
</tr>
<tr>
<td>Gabon Grid</td>
<td>Gauss-Kruger Zones 1 through 4 (DHDN). Versions of the Gauss-Kruger projection as used within Germany. See the Gauss Kruger topic.</td>
</tr>
<tr>
<td>Germany</td>
<td>Gauss-Kruger Zones 1 through 4 (DHDN). Versions of the Gauss-Kruger projection as used within Germany. See the Gauss Kruger topic.</td>
</tr>
<tr>
<td>Greek Grid</td>
<td>GS50 projection for the 50 States</td>
</tr>
<tr>
<td></td>
<td>GS50 projection for the 50 States (Spherical)</td>
</tr>
<tr>
<td>Guam</td>
<td>HGRS87</td>
</tr>
<tr>
<td>Irish National Grid</td>
<td></td>
</tr>
<tr>
<td>Israel Grid</td>
<td></td>
</tr>
<tr>
<td>Italy</td>
<td>Zone 1 (West) and Zone 2 (East).</td>
</tr>
<tr>
<td>Jordan Transverse Mercator</td>
<td></td>
</tr>
<tr>
<td>Korean Grid</td>
<td>Korea Central, Korea East and Korea West.</td>
</tr>
<tr>
<td>Lee Oblated</td>
<td>Stereographic projection for the Pacific Ocean</td>
</tr>
<tr>
<td>Lithuania</td>
<td></td>
</tr>
</tbody>
</table>
Luxembourg

Micronesia

Individual projections for Majuro Atoll (Marshall Islands), Palau (Caroline Islands), Ponape (Caroline Islands), Rota (Mariana Islands), Saipan (Mariana Islands), Truk Atoll (Caroline Islands), and Yap (Caroline Islands).

Miller Oblated

Stereographic projection for Europe and Africa

Modified-Stereographic Conformal projection for Alaska

Modified-Stereographic Conformal projection for Alaska (Spherical)

Modified-Stereographic Conformal projection for bounded United States

New Zealand Grid

New Zealand Grid, New Zealand Transverse Mercator, Old North Island Grid and Old South Island Grid.

Norway

Zones 1 through 8.

Oblique Mercator Laborde

Oblique Mercator Rosenmund

Oblique Mercator Spherical

Oman Grid

PDO Survey Datum 1993

Palestine Belt

Palestine Grid

Philippines Grid

Zones I through V

Portuguese Grid

Qatar Grid

Singapore Grid

Spanish Grid

State Plane (NAD 27), Feet


State Plane (NAD 83), Feet


State Plane (NAD 83), Meters

| Swedish Grid | Zones 1 through 6 |
| Swiss Grid | New Swiss Grid and Old Swiss Grid. |
| Transverse Mercator South Oriented | |
| Trinidad Grid | |
| United States | Specialized projections used within the United States, including: |
| | • ATnT V and H (AT&T) |
| | • Michigan GeoRef (Michigan) |
| | • Mississippi Transverse Mercator |
| | • Shackelford (Texas) |
| | • Teale (California) |
| | • Texas CMS / ACEA (Texas) |
| | • Texas CMS / LCC (Texas) |
| | • Texas SMS (Texas) |
**GS50 Projection**

A conformal projection designed to show the 50 states of the United States.

**Scale**

True along an irregular line approximately encompassing the regions of the 50 states of the United States. Varies less than 2 percent from true scale throughout the 50 states and adjacent bodies of water. This is less than one-fourth the variation in scale of the best standard projections.

**Distortion**

Shape and scale distortion very low for the 50 state region. Greater distortion for regions away from the 50 states so that the projection is unusable outside the 50 states and immediately adjacent regions.

**Usage**

Maps of the 50 states of the United States, especially maps for school children that show the true relative sizes and shapes of Alaska and the lower 48 states.

**Limitations**

Use only for US 50 state region.

**Origin**

Developed by J.P. Snyder of the U.S. Geological Survey in 1982. The **spherical** case uses a sphere for the Earth's ellipsoid.

The GS50 uses tenth-order complex-algebra polynomials to modify the Stereographic projection. Conformality is precise in these cases even though finite polynomial series are used.

**Options**

The coefficient choices are useful only for this design of a 50-State map.
State Plane Coordinate System
The State Plane Coordinate System is a set of many different projections organized by Zone. Transverse Mercator projections are the basis for the State Plane Coordinate System in U.S. States having predominately north-south extent. Lambert Conformal Conic projections are also used in many of the U.S. State Plane Coordinate System zones.

The State Plane Coordinate System projections are found under the National Grids heading in Manifold projections dialogs. There are three choices:

- **State Plane (NAD27, feet)**: State Plane Coordinate System used within the United States based on NAD27 datum. A mix of different projections and parameters, but mostly UTM-based. In feet.

- **State Plane (NAD83, feet)**: State Plane Coordinate System used within the United States based on NAD83 datum. A mix of different projections and parameters, but mostly UTM-based. In feet.

- **State Plane (NAD83, meters)**: State Plane Coordinate System used within the United States based on NAD83 datum. A mix of different projections and parameters, but mostly UTM-based. In meters.

Note that the NAD 27 version is provided in feet. Since some State Plane users of the NAD 83 version use either feet or meters the NAD 83 version is provided in both feet and meter based projections.

Limitations

The accuracy of Transverse Mercator projections quickly decreases from the central meridian. Therefore, it is strongly recommended to restrict the longitudinal extent of the projected region to +/- 10 degrees from the central meridian. [The US Army standard allows +/- 24 degrees from the central meridian].

This requirement is met within all State Plane zones that use Transverse Mercator projections. However, given modern computer technology these recommended limits result in ludicrously inaccurate maps. But then again, people seriously interested in exploiting the power of modern computers would not be using the State Plane system in the first place.

Because many state plane zones use Transverse Mercator projections they are subject to the same difficulties and limitations of the Universal Transverse Mercator (UTM) system, which also employs Transverse Mercator projections. It is very important for new users to realize that each State Plane Zone is a separate projection. Due to distortion effects induced by Transverse Mercator projections the State Plane system is highly unsuited for creating maps that combine more than more State Plane zone.

See the Universal Transverse Mercator (UTM) topic for a discussion of this and other problems with Transverse Mercator projections.

Comments

Like UTM or Gauss Kruger, the State Plane Coordinate System is a living fossil. It was created mostly as a reflection of the technological limitations of an earlier era. When used with skill as originally intended it still functions well in expert hands. The problem with the
State Plane system (as with UTM and Gauss Kruger) is that it is constantly misused in civil applications by inexpert users who do not realize the limitations built into the system.

**Standard**

**Latitude / Longitude Projection**

Not really a projection, but the graphical figure obtained by plotting latitude and longitude degree coordinates as if they were $Y$ and $X$ coordinates. The default "unprojection," it is also known as the geographical latitude / longitude projection.

Note that one latitude / longitude projection may be different from another because each such projection assumes that a particular datum is in use. While the WGS84 datum is almost universally assumed to the datum used, it is possible that some person assumed a different datum.

It is also possible to re-project latitude / longitude data into different datums, even including datums that include rotation factors. In such cases Manifold will faithfully reproject the data as direct taking into account datum rotation factors if applicable.

**Scale**

True in degrees along the Equator or vertically along any meridian.

**Distortion**

Considerable distortion away from the Equator due to horizontal increase in longitude degrees.

**Usage**

Used to save digital maps in unprojected form using some assumed datum.

**Azimuthal**

Azimuthal Equidistant
An azimuthal projection that is neither equal-area nor conformal. The main feature of this projection is that it shows distances and directions correctly from the central point.

**Scale**

All distances measured from the center are true. Distances not measured along radii from the center are not correct.

**Distortion**

The center of the projection is the only point without distortion. Directions from the center are true, except on some oblique and Equatorial ellipsoidal forms.

**Usage**

Used in the polar aspect for world maps and maps of polar hemispheres. Used in the oblique aspect for atlas maps of continents and world maps for aviation and radio use. The ellipsoidal oblique aspect is used for the plane coordinate projection system in approximate form for Guam and in nearly rigorous form for islands in Micronesia.

**Limitations**

Use only for a single hemisphere at a time

**Origin**

Known for many centuries in the polar aspect. It is believed the Egyptians used the polar aspect for star charts, but the oldest existing celestial map using this projection was prepared in 1426 by Conrad of Dyffenbach. The first known use for polar maps of the Earth was by Gerardus Mercator as insets on his 1569 world map, which introduced his famous cylindrical projection.
Donald
An azimuthal, two-point equidistant projection specially modified for the Earth ellipsoid and confined to the United States and southern Canada. Shows true distances, but not true azimuths, from either of two chosen points to any other point on the map.

Scale
True along a straight line from either of the two central points that define the projection. Distances are exact when measured from either one of two central points, at: 37 degrees 42 minutes 14.69 seconds North (37.7040806) latitude, 82 degrees 39 minutes 15.27 seconds West (-82.6542417) longitude in Floyd County, Kentucky and at 41 degrees 02 minutes 55.53 seconds North (41.0487583) latitude, 112 degrees 03 minutes 39.35 seconds West (-112.0609306) longitude in Webster County, Utah.

Distortion
No points free of distortion. Minimizes distance distortion in the lower 48 US states.

Usage
Used by telephone companies to establish long-distance rates in the United States and southern Canada.

Limitations
Use only for the lower 48 United States and Southern Canada.

Origin
Developed by Jay K. Donald of the American Telephone and Telegraph Company in 1956. Two-point equidistant projections were originally presented by Hans Maurer (1868-1945) of Germany in 1919 and by Charles F. Close (1865-1952) independently in 1921.
Double Stereographic

An azimuthal, conformal, polyconic (general) perspective projection that is visually similar to the ordinary Stereographic. Used most conveniently with a single hemisphere. This version of the Stereographic projection is called the "Double" Stereographic because it is really two projections: the surface of the utilized ellipsoid is first conformally projected to a sphere and then the sphere is stereographically projected to a plane. It therefore provides both a perspective view while also providing conformal mapping of the original ellipsoid onto a plane.

Scale

True only where the central latitude crosses the central meridian or, alternatively, along a circle concentric about the projection center (or a parallel on the polar aspect). Scale is constant along any circle having its center at the projection center, but scale increases rapidly with distance from the center within a hemisphere.

Distortion

Only the center or the circle of true scale (if not the center) is free from all distortion. Areas grow greater the farther from the center, albeit in a conformal manner.

Usage

Similar to that of the ordinary Stereographic. Commonly used in the polar aspect for topographic maps of polar regions. The Equatorial aspect was used regularly for maps of the Eastern and Western hemispheres in the 17th and 18th centuries. Oblique aspects are used to show paths of solar eclipses.

Recommended for conformal mapping of regions approximately circular in extent. For example, in 1997 it was used as the standard projection for the Fermilab Main Injector project, which involved injecting protons and antiprotons into a large, circular Tevatron ring. By centering the Double Stereographic projection at the center of the Tevatron, scale throughout the circular tunnels hosting the Tevatron could be kept accurate.

Limitations

The Double Stereographic projection must not be used to map the entire world’s surface at once: at least the point directly opposite to the projection origin must be excluded. This
limitation arises because the Double Stereographic projection maps the point opposite the projection origin to infinity, causing numeric overflows. For example, if the North Pole is used as the projection origin, the South Pole and region immediately about the South Pole should not be included in the map.

**Origin**

Apparently developed in polar aspect by Egyptians and Greeks by the 2nd Century BC

**Options**

Specify the latitude origin and longitude origin to center the map projection to the area to be mapped. Specifying a non-Equatorial or non-polar origin causes an oblique projection.
Gnomonic

An azimuthal projection that is a perspective projection and is neither conformal nor equal-area. Also called the Gnomic or Central projection and was known as the horologium (meaning "sundial" or "clock") in earlier times.

Scale

Scale true at the center only. Directions from the center are true.

Distortion

No distortion at the center only. Distortion and scale rapidly increase away from the center.

Usage

The only useful feature of the Gnomonic projection is that all great-circle arcs project as straight lines on this projection. The scale is badly distorted along such a plotted great circle, but the route is precise.

Except at the center, distortion of shape, area and scale on the Gnomonic projection is so great that it has seldom been used for atlas maps. Generally, the projection is used for plotting great-circle paths.

Several sets of star maps from the late 18th century and some terrestrial maps of 1803 used this projection with the sphere projected onto the six faces of a tangent cube. Also used from the mid-16th to the mid-20th centuries to project the terrestrial globe onto the faces of other polyhedra.

Limitations

Use only for a single hemisphere.

Origin

Used by Thales (636? - 546? BC) of Miletus for star maps.
Limiting Forms

Used in Spherical form only.
Lambert Azimuthal Equal Area

An azimuthal, equal area, nonperspective projection.

**Scale**

True only at the center in all directions. Decreases with distance from the center along radii. Increases with distance from the center in a direction perpendicular to radii.

**Distortion**

Only the center is free from distortion. Distortion is moderate for one hemisphere but becomes extreme for a map of the entire Earth.

**Usage**

Frequently used in the polar aspect in atlases for maps of polar regions and of Northern and Southern Hemispheres. The Equatorial aspect is commonly used for atlas maps of the Eastern and Western Hemispheres. The oblique aspect is used for atlas maps of continents and oceans. The Equatorial and oblique aspects are used by the U.S. Geological Survey in cooperation with others for maps of the Circum-Pacific Map Project.

Recommended for equal area maps of regions approximately circular in extent.

**Limitations**

Use only for a single hemisphere.

**Origin**

Presented by Johann Heinrich Lambert (1728-1777) of Alsace in 1772. Also known as Lorgna (for the polar aspect), Zenithal Equal-Area or Zenithal Equivalent.
Johann Heinrich Lambert

Lambert served at the Berlin Academy of Sciences, a contemporary of Euler and Lagrange. An extraordinary mathematician, he is best known as the first to prove rigorously that $\pi$ is irrational.

Options

Specify the **first standard parallel** and **longitude origin** to center the map projection to the area to be mapped. Specifying a non-Equatorial or non-polar origin causes an **oblique** projection.
Modified Stereographic
A modified azimuthal projection that is conformal. One of an endless possible series of
conformal projections using the Driencourt and Laborde algebraic transformations.

Scale

Scale is true along irregular lines, but the map is usually designed to minimize scale
variation throughout a selected region.

Distortion

Low, but continuous distortion within the target region. Profound distortion outside the
intended range of the map.

Usage

Used for maps of continents in the Easter Hemisphere, for the Pacific Ocean and for maps
of Alaska and the 50 United States. Provides extremely constant scale over regions as
large as the conterminous US. The GS-50 version of this projection minimizes scale errors
through all 50 United States including Alaska and Hawaii as well as island possessions.

Limitations

Use only for a single hemisphere.

Origin

Based on the Driencourt and Laborde complex algebraic series that satisfies the Cauchy-
Riemann equations that define the conformal transformation of one surface onto another
surface. Laborde applied this transformation to the mapping of Madagascar in 1928.

Options

Manifold provides a standard set of the most commonly used forms of projections based
on Driencourt and Laborde terms.
Orthographic

The "view from space" projection: An azimuthal, perspective projection that is neither conformal nor equal area. Range is no more than one hemisphere at a time. This is the default Manifold projection for new drawings, images and labels components.

Scale

True at the center and along any circle having its center at the projection center but only in the direction of the circumference of the circle. Scale decreases with distance from the center.

Distortion

Only the center is free from distortion, which increases rapidly away from the center. Distortion is extreme near the edge of the hemisphere.

Usage

Pictorial views of the Earth, resembling those seen from space. This is a perspective projection of the globe onto a tangent plane from an infinite distance (i.e., orthogonally); thus, the map has the look of a globe. The Orthographic projection is used by default within Manifold for images and drawings that are not otherwise georegistered.

Limitations

Use only for a single hemisphere. Defined only for a sphere, specifically a sphere that utilizes the same major axis as the WGS84 datum, 6378137. Note that although WGS84 appears as a datum for Orthographic projection, the flattening of the WGS84 ellipsoid is ignored and only the major axis is used to define the sphere. Note that this is different than the explicitly enumerated Sphere datum, which utilizes a major axis of 6370997. Use the Stereographic projection instead of Orthographic if non-spherical datums are to be utilized.

Origin

Apparently developed by Egyptians and Greeks by the 2nd Century BC
Options

Specify the center of the projection by setting **latitude origin** and **longitude origin**. Specifying a non-Equatorial or non-polar origin causes an **oblique** projection. The **Clip coordinates** check box clips invisible parts of objects, such as countries located on the other side of the Earth from the point of view of the projection. Note that **Clip coordinates** is a destructive change: parts of objects extending beyond the projection horizon will be permanently trimmed.

The Southern Hemisphere view above is created using a latitude origin of -90 and a longitude origin of 0.

The Eastern Hemisphere view above is created using a latitude origin of 0 and a longitude origin of 90.

The Western Hemisphere view above is created using a latitude origin of 0 and a longitude origin of -90.
If more than one hemisphere is displayed, countries will be "wrapped" from the invisible side of the world and displayed anyway in mirror image.

Above is an Orthographic projection centered on latitude 68 North longitude -70. The original map included areas and a graticule for just the Northern Hemisphere. If zoomed far into the latitude and longitude origin we would see essentially zero distortion.

**Using Clip Coordinates**

The *Clip coordinates* checkbox in the *Edit - Change Projection* dialog tells Manifold to cut objects so that they do not "wrap" around the Orthographic coordinate system.

If we import the *World_eg.mfd* sample drawing and use *Edit - Change Projection* to re-project it into Orthographic centered on the default 0, 0 origin we will see that some areas, such as Australia and New Zealand are "wrapped" around the edge of the Orthographic system. This effect arises because the Orthographic projection is not intended to deal with more than one hemisphere's worth of data at a time.

To avoid this effect we have two choices:

- Edit the drawing in advance so that only areas in the hemisphere of interest exist. This is the method used to create the screenshots in the previous section of this topic.
- Check the *Clip coordinates* box and let Manifold "trim" objects to only those parts that are visible in the centered hemisphere. This works well in many situations but can be very slow to compute with large drawings since it is an extremely
computationally expensive process (as will be seen from the operations reported in the progress dialogs it shows).

The illustration above shows the World_eg sample drawing after projection to Orthographic using Clip Coordinates. Note there is no overlap.

The Clip coordinates box is a convenience, not an exact cartographic instrument. It works by clipping objects using a clipping rectangle. If after using Clip coordinates in an Orthographic projection we re-project the World_eg drawing back into Latitude / Longitude we will see the areas have been clipped to fit inside a bounding box.

Using straight lines to clip objects at the Orthographic horizon is an imperfect approximation (it leads to a "lumpy" horizon sometimes at the edge of the Earth), but it is reasonably fast to compute.

**Technical Note: Why is Clip Coordinates so Slow?**

Users will soon learn that checking the Clip coordinates box changes the time required for a re-projection from nearly instantaneous to usually taking minutes or even hours although sometimes even with the Clip coordinates box checked the re-projection is still fast. Why is that?

Before Manifold System release 7.00 the Clip coordinates algorithms were relatively imprecise, trading accuracy away in order to gain speed. A side effect of that reduced accuracy was that in certain cases of very large data the error conditions accumulating from reduced accuracy would cause data to grow without end and result in a "no memory" error and a failed re-projection.

From release 7.00 the new Clip coordinates algorithms are much more accurate, using precision over 100 times greater than previously. In addition, the algorithms have been adjusted so that no matter how large the data set involved will Manifold run out of memory. The price of these changes for greatly improved accuracy and ability to handle very large data sets is that performance in all cases is really miserable. It is expected that future releases will introduce algorithmic improvements that will deliver performance gains.
In the meantime, users have two strategies to increase the performance when using the **Clip coordinates** box:

- Plan your work to schedule any re-projections requiring **Clip coordinates** for the end of your work day, so that the re-projection can be set cooking while you are away from the project and won't notice how long it takes. Plan big re-projections requiring **Clip coordinates** for the end of the day Friday when Manifold will have the entire weekend to crank away.
- Do re-projections only where the target center latitude is **zero**. A side effect of the algorithms used is that when the target center latitude is zero, the clipping areas used internally within the algorithms become much simpler and easier to compute and so the process goes much faster. As a result, if re-projecting to Orthographic where the target center latitude is zero, even with **Clip coordinates** checked the re-projection will still go reasonably fast.

**Notes on Default Use of Orthographic Projection**

Manifold uses Orthographic projection as a default projection when nothing is known about the geographic context of a component.

Geographic components imported into Manifold from geographically aware formats will automatically use whatever geographic context or projection is defined in the source file(s). Components (such as images and CAD drawings) imported from non-geographic formats will be imported as if they were in a meter-based Orthographic projection. Such components may acquire a geographic context at some point by georegistration but initially they are usually created in meter-based or pixel-based coordinate systems.

It makes sense to consider such components as being in Orthographic projection. Considering the small size of CAD drawings or images relative to the size of an entire Earth hemisphere, there is an essentially perfect correlation between the flat, Euclidean coordinates of the drawing or image space and the effectively flat, Euclidean coordinates of the very central portion of an Orthographic projection.

This convention of considering all abstract coordinate CAD drawings and pixel coordinate images to be emplaced within an Orthographic projection provides three main benefits:

- First, it allows standard dialogs for both geographic and non-geographic components because all components within Manifold are treated as geographic components.
- Second, it provides a simple way of dealing with geographic components imported from formats that do not store projection information: one imports the projection in a default way and then changes the component's projection properties to the required values.
- Third, many images used in GIS work are overhead images that may be assumed to be in Orthographic projection as a practical approximation for most purposes. This makes it relatively easy to georegister such images in a simple way.

See the Manually Georegister an Image topic for an example.
Often incorrectly spelled as the "Pierce" projection, this projection is a conformal projection also known as a quincuncial projection. It is a transverse case of the Guyou projection.

**Caution:** This is a "one-way" projection in terms of accuracy in that once data is projected into the Peirce coordinate system it cannot be re-projected to other systems without some loss of precision. The definition of the projection in common use does not include formulae for the inverse transformation, so projecting from Peirce to latitude/longitude uses iterative approximations and thus can lose precision.

**Scale**

Not true anywhere. Scale is especially false in the corners of the projection, where scale is elongated, and at the poles, where scale is compressed.

**Distortion**

The projection is conformal except at the four "corners" of the Equator. Conformality (and therefore local angle preservation) fails at the corners.

**Usage**

Useful for showing polar regions. In its complete form (the entire Earth), may be tiled infinitely to create a display that allows any point on Earth to be viewed in a direct line from any other point.

**Limiting Forms**

The Guyou and Peirce projections are transverse cases of each other. The Guyou is the equatorial aspect (hence the classification as a psuedocylindrical projection within Manifold) and the Peirce is the polar aspect (hence the classification as an azimuthal projection within Manifold).
The term "quinuncial" refers to a presentation using a whole-Earth form of this projection into a square consisting of a central diamond square and four corner one-fourth squares, thus having five regions (Latin for five is quinque).

Options

Specify the center of the projection by setting latitude origin and longitude origin.

Origin

Invented by Charles Sanders Peirce in 1879 while he was at the U.S. Coast and Geodetic Survey.

Charles Sanders Peirce (1839-1914)

A classic example of a very bright individual who dabbled in many areas but left virtually no enduring mark in history. Peirce was a troublesome employee at the U.S. Coast and Geodetic Survey until eventually resigning in 1891. He invented a philosophical approach called Pragmatism he hoped would provide a counterpoint to Kant, invented a method of obtaining acetylene, set forth a means for defining the meter by a wavelength of light, was fond of presenting logic using "semiotics" (a theory of signs), observed occultations, obtained a degree in Chemistry at Harvard in 1863 and died mired in poverty and illness in 1914 after being unable to find steady employment after leaving the Survey.

Peirce continues to have his fans, who create websites reprinting his philosophical gems such as: "Thought is what it is only by virtue of its addressing a future thought which is in its value as thought identical with it, though more developed. In this way, the existence of thought now depends on what is to be hereafter; so that it has only a potential existence, dependent on the future thought of the community." ...Thus speaks a pragmatist!

The Peirce quincuncial projection seems typically Peirce in that its use of elliptical integrals for computation achieves a result through fearsomely complex means when simpler methods would seem to serve as well.

When contemplating Peirce's life one wonders how it is that accidents of history have elevated some would-be philosophers to superstardom while others have been consigned to the dustbin of history. For example, but for an accident of utopian popularity, Marx, too, could have ended up as yet another unknown kook beavering away at writings that made sense only to him.
Stereographic

An azimuthal, conformal, polyconic (general) perspective projection. Used most conveniently with a single hemisphere. Illustrated centered on -90 degrees longitude and 0 degrees latitude.

The Polar Stereographic form of this projection (polar aspect) is provided as a separate projection within the azimuthal projections. It is illustrated above.

Scale

True only where the central latitude crosses the central meridian or, alternatively, along a circle concentric about the projection center (or a parallel on the polar aspect). Scale is constant along any circle having its center at the projection center, but scale increases moderately with distance from the center within a hemisphere.

Distortion

Only the center or the circle of true scale (if not the center) is free from all distortion. Areas grow greater the farther from the center, albeit in a conformal manner.

Usage
Commonly used in the polar aspect for topographic maps of polar regions. The Equatorial aspect was used regularly for maps of the Eastern and Western hemispheres in the 17th and 18th centuries. Oblique aspects are used to show paths of solar eclipses.

Recommended for conformal mapping of regions approximately circular in extent.

**Limitations**

The Stereographic projection must not be used to map the entire world's surface at once: at least the point directly opposite to the projection origin must be excluded. This limitation arises because the Stereographic projection maps the point opposite the projection origin to infinity, causing numeric overflows. For example, if the North Pole is used as the projection origin, the South Pole and region immediately about the South Pole should not be included in the map.

**Origin**

Apparently developed in polar aspect by Egyptians and Greeks by the 2nd Century BC

**Options**

Specify the latitude origin and longitude origin to center the map projection to the area to be mapped. Specifying a non-Equatorial or non-polar origin causes an oblique projection.

**Tech Tip**

The top illustration, showing a partially clipped Africa and Europe, was created using a trick to clip Europe and Africa. A graticule from 0 longitude to 90 longitude and from -80 latitude to 80 latitude was created as lines. Areas were then created within the graticule and unioned together to form one large area. This was used with the Clip with (Subtract) transform to clip those portions of Europe and Africa within the graticule area. Other areas in Europe, Africa and Asia were simply selected and deleted free hand. The remaining, unclipped portions of Europe and Africa fit neatly within the 0 to -90 longitude graticule seen in the illustration.
Tilted Perspective

A true "view from space" projection. Whenever the Earth is photographed from space, the camera records the view as a perspective projection. If the camera precisely faces the center of the Earth the projection is Vertical Perspective. If the camera does not precisely face the center of the Earth the projection is a Tilted Perspective.

Perspective projections are neither conformal nor equal area. They were known by Greeks and Egyptians in ancient times. Vertical Perspective projections are azimuthal; Tilted Perspective projections are not.

Scale

Scale is accurate only at the center for Vertical Perspective. Directions from the center are true for the Vertical Perspective for the sphere and for the polar ellipsoidal form.

Distortion

No distortion at the center for Vertical Perspective. Considerable distortion near the edges of the projection and throughout Tilted Perspective projections.

Usage

Used for pictorial views of the Earth as seen from space. Vertical Perspective is used for weather maps issued by the U.S. National Weather service as seen from geosynchronous satellites near the Equatorial plane. Tilted perspective in the ellipsoidal form is used for Space Shuttle Large Format Camera images and other photographs.

Limitations

Use only for a single hemisphere.

Origin

Known by Greeks and Egyptians in ancient times.

Limiting Forms
Orthographic, Stereographic and Gnomonic projections are special forms of the Vertical Perspective.
Vertical Perspective

A true "view from space" projection. Whenever the Earth is photographed from space, the camera records the view as a perspective projection. If the camera precisely faces the center of the Earth the projection is Vertical Perspective. If the camera does not precisely face the center of the Earth the projection is a Tilted Perspective.

Perspective projections are neither conformal nor equal area. They were known by Greeks and Egyptians in ancient times. Vertical Perspective projections are azimuthal; Tilted Perspective projections are not.

Scale

Scale is accurate only at the center for Vertical Perspective. Directions from the center are true for the Vertical Perspective for the sphere and for the polar ellipsoidal form.

Distortion

No distortion at the center for Vertical Perspective. Considerable distortion near the edges of the projection and throughout Tilted Perspective projections.

Usage

Used for pictorial views of the Earth as seen from space. Vertical Perspective is used for weather maps issued by the U.S. National Weather service as seen from geosynchronous satellites near the Equatorial! plane. Tilted perspective in the ellipsoidal form is used for Space Shuttle Large Format Camera images and other photographs.

Limitations

Use only for a single hemisphere.

Origin

Known by Greeks and Egyptians in ancient times.

Limiting Forms
Orthographic, Stereographic and Gnomonic projections are special forms of the Vertical Perspective.

Conic

Albers Conical Equal Area

A conic, equal area projection.

Scale

True along the one or two chosen standard parallels, which are usually but not necessarily specified on the same side of the Equator. As a rule of thumb, these parallels can be placed at one-sixth and five-sixths of the range of latitudes, but there are more refined means of selection.

Scale is constant along any given parallel. The scale factor at any given point along the meridian is the reciprocal of that along the parallel, to preserve area.

Distortion

Free of angular and scale distortion only along the one or two standard parallels. Distortion is constant along any given parallel.

Usage

Frequently used in the ellipsoidal form for maps of the United States in the National Atlas of the United States, for thematic maps, and for world atlases. Also used and recommended for equal-area maps of regions that are predominantly east-west in extent.

Limitations

Normally used only for a single hemisphere.

Origin

Presented by Heinrich Christain Albers (1773-1833) of Germany in 1805.

Limiting Forms
Polar Lambert Azimuthal Equal-Area projection: if a pole is made the single standard parallel. The cone of projection thereby becomes a plane.

Lambert Equal-Area Conic projection: if the pole and another parallel are made the two standard parallels.

Lambert Cylindrical Equal-Area projection: if the Equator is the single standard parallel. The cone of projection thereby becomes a cylinder.

Behrmann or other cylindrical equal-area projections: if the two standard parallels are symmetrically placed north and south of the Equator.

**Options**

Specify the **first standard parallel** and **second standard parallel** to tailor the projection to the area to be mapped.

Specify the **latitude origin** and **longitude origin** to center the map projection to the area to be mapped. Specifying a non-Equatorial origin causes an **oblique** projection.
Bipolar Oblique Conformal Conic

A conformal projection that consists of two oblique conic projections, side-by-side, but with poles 104 degrees apart.

**Scale**

Scale is true along two standard transformed parallels on each conic projection. Neither of these lines follows any geographical meridian or parallel.

**Distortion**

Very small deviation from conformality, where the two conic projections join. However, potentially very large distortion outside the intended region. When creating graticules for use with this distortion the graticule must be trimmed by selecting portions of the graticule and deleting those portions that otherwise are projected in unpleasant ways. Note how the graticule shown in the illustration above has been trimmed.

**Usage**

Used only in spherical form for maps of the Americas. Used by the AGS for maps of North and South America, by USGS for the North American portion of the Geologic Map of 1965, the Basement Map of 1967 and the USGS Geothermal Map and Metallogenic Map.

**Limitations**

Use only for maps of the Americas.

**Origin**

Developed by O.M. Miller and William A. Briesemeister and presented in 1941 and designed specifically for a map of North and South American constructed in several sheets by the American Geographical Society (AGS).
Bonne

A psuedoconical, equal-area projection. Central meridian is a straight line. All other meridians are complex curves.

Scale

Scale is true along the central meridian and along all parallels.

Distortion

No distortion along the central meridian and along the standard parallel.

Usage

Used for atlas maps of continents and for topographic mapping of some countries.

Limitations

Use only for a single hemisphere.

Origin

Used considerably by Rigobert Bonne (1727-1795) in the mid-18th century, but developed by others during the early 16th century.

Limiting Forms

Sinusoidal projection is the Equatorial limiting form of the Bonne projection. The polar limiting form is a cordiform (heart-shaped) world map devised by Johannes Stabius and used widely by Johannes Werner in about 1514 and is known as the Werner projection. The Werner projection was used by Mercator and Ortelius in the late 16th century for maps of Asia and Africa. The Bonne projection has less distortion because its projection center is at the center of the region being mapped instead of at the pole.
Equidistant Conic

A conic projection with equally spaced parallels. Neither conformal nor equal area. Also known as the Simple Conic or Conic projection.

Scale

True along each meridian and along one or two chosen standard parallels, usually but not necessarily on the same side of the Equator. As a rule of thumb, these parallels can be placed at one-sixth and five-sixths of the range of latitudes, but there are more refined means of selection.

Scale is constant along any given parallel.

Distortion

Free of angular and scale distortion only along the one or two standard parallels. Distortion is constant along any given parallel. Compromise in distortion between equal-area and conformal conic projections.

Usage

The most common projection in atlases for small countries. Also used by the Soviet Union for mapping that nation.

Limitations

Use only for a single hemisphere.

Origin

Rudimentary forms developed by Claudius Ptolemy (about A.D. 100). Improvements by Johannes Ryusch in 1508, Gerardus Mercator in the late 16th century, and Nicholas de l’Isle in 1745.

Limiting Forms

Polar Azimuthal Equidistant projection: if a pole is made the single standard parallel. The cone of projection thereby becomes a plane.
Plate Carree projection: if the Equator is the single standard parallel. The cone of projection thereby becomes a cylinder.

Equirectangular (Cylindrical) projection: if the two standard parallels are symmetrically placed north and south of the Equator.

**Options**

Specify the **first standard parallel** and **second standard parallel** to tailor the projection to the area to be mapped.

Specify the **latitude origin** and **longitude origin** to center the map projection to the area to be mapped. Specifying a non-Equatorial origin causes an **oblique** projection.
Krovak Oblique Conformal Conic
An oblique variation of the Lambert Conformal Conic projection. The standard Lambert Conformal Conic projection places the axis of the projection cone coincident with the minor axis of the Earth ellipsoid. That is, the axis of the cone is normal to the Earth ellipsoid at the pole.

The Krovak projection places the axis of the cone normal to the Earth ellipsoid at some other location and so the line of the axis of the cone intersects the minor axis at some defined angle.

Scale
True only along the pseudo standard parallel.

Distortion
Free of distortion only along the pseudo standard parallel.

Usage
Used in the Czech Republic and Slovakia

Limitations
Use only for a single hemisphere centered on the given central latitude and longitude.

Origin
Created by Professor Josef Krovak of Czechoslovakia in 1922 for tax and topographic maps for the Czechoslovakian geodetic service.

Limiting Forms
The Lambert Conformal Conic projection when the axis is normal to the pole.

Options
Specify the Center Latitude and Center Longitude (normally centered on the area of interest). The Azimuth of the center line is the true azimuth of the center line passing through the center of the projection. This is also known as the "co-latitude" of the cone axis at point of intersection with the ellipsoid.

The Latitude of pseudo standard parallel is the latitude of the radius arc on the cone that is true to scale.
Lambert Conformal Conic

A conformal conic projection. Also known as the Conic Orthomorphic projection. A USGS and Manifold favorite for maps of the US. The **Lambert Conformal Conic (Single Parallel)** choice is a simplified version that uses only one parallel.

**Scale**

True along one or two chosen standard parallels, usually but not necessarily on the same side of the Equator. As a rule of thumb, these parallels can be placed at one-sixth and five-sixths of the range of latitudes, but there are more refined means of selection.

Scale is constant along any given parallel and is the same in all directions at a given point.

**Distortion**

Free of distortion only along the one or two standard parallels. Distortion is constant along any given parallel. Conformal everywhere except at the poles.

**Usage**

Extensively used in ellipsoidal form for large-scale mapping of regions of predominantly east-west extent, including topographic quadrangles (1:24,000 and 1:62,500 scale) for many of the U.S. State Plane Coordinate system zones, many maps in the International Map of the World (1:1,000,000 scale) series, the U.S. State Base Maps (1:500,000 scale), and topographic mapping in many other nations.

Also used for atlas maps of some countries. Recommended for conformal mapping of regions of predominantly east-west extent, such as the US or Russia.

**Limitations**

Normally used only for a single hemisphere.

**Origin**

Presented by Johann Heinrich Lambert (1728-1777) of Alsace in 1772.

**Limiting Forms**

...
Polar Stereographic projection: if a pole is made the single standard parallel. The cone of projection thereby becomes a plane.

Mercator projection: if the Equator is the single standard parallel or if two standard parallels are symmetrically placed north and south of the Equator. The cone of projection thereby becomes a cylinder.

**Options**

Specify the first standard parallel and second standard parallel to tailor the projection to the area to be mapped.

Specify the latitude origin and longitude origin to center the map projection to the area to be mapped. Specifying a non-Equatorial or non-polar origin causes an oblique projection.

The Lambert Conformal Conic (Single Parallel) choice is a simplified version that uses only one parallel, the central latitude. This has slightly more distortion than the full version since the cone of projection is tangent to the Earth at the one parallel.

**Example**

For maps of the United States, reasonable figures are 38 for the Center Latitude and -100 for the Center Longitude. This is a spot approximately in the center of the lower 48 states of the United States.

Use 32 for the 1st Standard Latitude and 44 for the 2nd Standard Latitude when using the full Lambert Conformal Conic version.

These are not hard and fast numbers. One may easily use 40 for the Center Latitude and 96 for the Center Longitude, which is probably closer to what most people would consider the “center” of the US. The important thing is to pick reasonable numbers that will be consistently used throughout one’s mapping, so that one’s projected maps using Lambert Conformal Conic to show the US will likely be the same. We use -100 because it is an easy number to remember, and Latitude 38 because it is the easily remembered name of a cool sailing journal on the West Coast (see www.latitude38.com).
Modified Polyconic

A pseudoconical projection that is neither conformal nor equal area. Based on the Polyconic, but differs from the ordinary Polyconic in two principal features: All meridians are straight, and there are two meridians that are made true to scale.

Scale

Meridians true to scale are 2 degrees east and west of the central meridian on sheets between latitudes 60 North and South, 4 degrees E / W between latitudes 60 and 76 N and S, and 8 degrees E / W between latitudes 76 and 84 N and S.

Distortion

Adjacent map sheets fit exactly together not only North to South but also East to West.

Usage

Used as the basis for the 1:1,000,000-scale International Map of the World (IMW) Series.

Limitations

Normally used only for a single hemisphere.

Origin

Devised by Lallemand of France and in 1909 adopted by the International Map Committee in London as the basis for the IMW. In 1962, a UN conference on the IMW adopted the Lambert Conformal Conic and Polar Stereographic projections to replace the Modified Polyconic.

Options

Specify the center longitude to center the map. The first standard latitude and second standard latitude should be approximately one fifth up from the bottom of the region covered and one fifth down from the top of the region covered. The true-scale longitude is usually the same as the center longitude.
Polyconic

A pseudoconical projection that is neither conformal nor equal area. Often called the American Polyconic in Europe. The central meridian and Equator are straight lines. All other meridians are complex curves.

Scale

True to scale along the central meridian. Each parallel is true to scale.

Distortion

Free of distortion along the central meridian and near the central meridian distortion is very small.

Usage

Used almost exclusively in slightly modified form for large-scale mapping in the United States until the 1950's.

Limitations

Normally used only for a single hemisphere.

Origin

Apparently originated about 1820 by Ferdinand Rudolph Hassler. Hassler was born in Switzerland in 1770, arrived in the US in 1850 and became the first Superintendent of the U.S. Coast Survey. Hassler died in Philadelphia in 1843 as result of exposure after a fall, trying to save his instruments in a severe wind and hail storm.

Cylindrical

Cassini
A cylindrical projection that is neither conformal nor equal-area.

Scale

True along central meridian and along lines perpendicular to central meridian. Scale is constant but not true along lines parallel to central meridian on spherical form, nearly so for ellipsoid.

Distortion

There is no distortion along the central meridian if it is maintained at true scale (the usual case).

Usage

Used for topographic mapping formerly in England and currently in a few other countries. Although the Cassini projection has been largely replaced by the Transverse Mercator, it is still in limited use outside the United States and was one of the major topographic mapping projections until the early 20th Century.

Origin

Devised by Cesar Francois Cassini de Thury (1714 - 1784) in 1745 for the survey of France. C.F. Cassini was the grandson of Jean Dominque Cassini, the famous Italian-born astronomer who changed his name from Giovanni Domenico after being hired in 1669 for astronomical research in Paris. J.D. Cassini began the survey of France and C.F. Cassini was the third of four generations involved in this project, the first detailed survey of a nation.
Cylindrical Equal Area

A cylindrical projection that is equal-area. Also known as the Lambert Cylindrical Equal Area projection. The projection upon which the even more distorting Gall projection is based.

Scale

No distortion of scale and shape at the standard parallels of the normal aspect but great distortion of shape and scale otherwise.

Distortion

Substantial shape and scale distortion near points 90 degrees from central line resulting in vertical exaggeration of Equatorial regions with compression of regions in middle latitudes and extreme vertical aspect compression in higher latitudes.

Usage

Most frequently used as a textbook example of the most easily constructed equal-area projection. Even with various proposed modifications (such as the Gall) distortions are so great that there has been little use of any of the forms for world maps by professional cartographers, many of whom have strongly criticized the intensive promotion in the non-cartographic community which has accompanied the political promotion of the re-named Gall modification (the infamous Peters projection).

Despite the shape distortion in some portions of a world map, this projection is well suited for equal-area mapping of regions which are predominantly north-south in extent, which have an oblique central line, or which lie near the Equator.

Origin

Devised by Johann Heinrich Lambert in 1772 and is the fourth of seven projections invented by him. Lambert described the transverse aspect which has very rarely been used. Even the normal aspect has been rarely used. A series of modifications were proposed by Gall in 1855, Behrmann in 1910 and others. Very similar projections were offered by Trystan Edwards of England in 1953 and a re-named copy of the Gall by Arno Peters of Germany in 1967. They were presented as revolutionary and original concepts rather than as modifications or copies of the prior projections.
Cylindrical Equidistant

A cylindrical projection that is neither equal-area nor conformal. One of the simplest of all map projections to construct and one of the oldest. Also known as the Rectangular, La Carte Parallelogrammatique, Die Rechteckige Platkarte and Equirectangular.

Scale

True to scale along the standard parallel.

Distortion

Considerable distortion away from standard parallel.

Usage

Used by USGS for index maps of the conterminous US with insets of Alaska, Hawaii and various islands on the same projection.

Origin

Probably originated with Erastosthenes (275? - 195? BC), the scientist and geographer known for his fairly accurate measure of the size of the Earth. Claudius Ptolemy credited Marinus of Tyre with the invention in AD 100.
Gall

A cylindrical projection that is equal-area. Very similar to the Cylindrical Equal Area projection (also known as the Lambert Cylindrical Equal Area projection) but with standard parallels at 45 degrees North and South.

Scale

No distortion of scale and shape at the standard parallels of the normal aspect but great distortion of shape and scale otherwise.

Distortion

Substantial shape and scale distortion near points 90 degrees from central line resulting in vertical exaggeration of Equatorial regions with compression of regions in middle latitudes and extreme vertical aspect compression in higher latitudes. Greater vertical aspect (taller) than the Cylindrical Equal Area projection upon which it is based.

Usage

Used mainly for political propaganda intended to maximize the apparent size of nations in Africa. Rarely used in professional cartography due to shape distortion. Usable for practical purposes only near the Equator.

Origin

Devised by James Gall in 1855 as a modification of the 1772 projection by Johann Heinrich Lambert.

Propaganda in Action: The "Peters Projection"

While cartographic technology, like many technologies, is of course routinely misused to lie to people for political ends, it is only on comparatively rare occasions through the centuries that political assaults have been launched against specific technologies within cartography itself. Such is the case with the "Peters projection" in modern times, when the twin hydra of political correctness and technical ignorance have combined to enmesh Gall's work within a contemporary propaganda movement. In an ironic twist, the misuse of cartography by this movement was advanced by an attack on the Mercator projection, an evil modern echo of the religious / political attack 450 years ago on Mercator himself.
Despite profound shape distortion and other defects, the Gall projection was adopted by Arno Peters of Germany in 1967, who falsely claimed the invention of the projection as his own. The "Peters projection" is, in fact, identical to the Gall except for an insignificant difference in standard parallels.

A student of political propaganda, Peters continued to claim his "invention" was distinctly different from the Gall projection even after it was conclusively demonstrated by professional cartographers that the so-called "Peters" projection was indeed the Gall. It is thus difficult to believe that Peters independently re-invented this projection as opposed to willfully plagiarizing the Gall, even if one is so credulous to think that Peters as an avid technical critic of the Mercator and other cylindrical projections could have somehow escaped noticing the vast literature describing the Lambert, the Gall and other similar projections. Like many evident plagiarists, Peters would have us believe that he managed to read the books he researched for "his" work while somehow honestly not noticing those parts he claims as his own.

Even after repudiation of their assertions by professional cartographers, both Peters and his political supporters continued to promote this "invention" through false claims. For example, the projection is claimed to be uniquely equal-area when there are numerous other projections, such as the Lambert Cylindrical Equal Area upon which it is based, that are also equal-area. It is claimed to be free of shape distortion when in fact it highly distorts shapes. It is claimed to preserve scale throughout the projection when in fact scale is true only along the two standard parallels and otherwise highly distorted throughout the projection.

Perhaps most telling of the political objectives of Arno Peters and his followers, the main claim for the projection is that it is more even-handed in presenting the true, relative size of Third World nations than are other projections, in particular the Mercator. But even this claim is false since any equal area projection will show the true areas of all nations, and numerous other projections do a better job of showing a more "realistic" apparent shape and size of different nations, including Third World nations, as they would be seen, say, to a politically-unbiased observer hovering above our planet. But then the very even-handedness of such projections works against them in service of the propaganda missions for which the "Peters" is deployed.

In fact, the Gall projection was apparently adopted by Arno Peters precisely because it distorts the apparent size of nations to overemphasize those nations favored by Arno Peters and his followers. While Peters criticized for apparent distortion the Mercator projection that is widely used for navigation worldwide, he and his followers falsely misrepresented his own use of the Gall projection as being free of distortion.

The Gall (and hence the "Peters" projection) greatly exaggerates the apparent size of equatorial countries through vertical scale exaggeration while compressing regions away from the Equator. It therefore makes First World countries in North America and Europe look smaller in comparison to Africa, which becomes enormous. This is, of course, a useful trick for political propaganda seeking to make Africa look larger and First World countries look smaller.

Note, by the way, that criticism by Peters supporters of the Mercator projection as motivated by political incorrectness is somewhat off the mark for those familiar with the technical characteristics of Mercator's projection and the history of its adoption.

As is well known to cartographers, the Mercator projection's wide adoption had nothing to do with any political agenda to minimize the geographic role of Third World countries. The Mercator won popularity first because it is easy to construct and second (and most importantly) because it shows rhumb lines as straight lines, a characteristic that is
extremely useful for navigation. The Mercator is also highly useful for conformal mapping of regions predominantly bordering the Equator, which is one reason it is often used by development and relief organizations wishing to employ effective tools in their assistance to equatorial Third World countries.

Keeping in mind that until the very recent, widespread dispersion of computers the cost of preparing virtually any cartographic presentation was very high, it made sense (especially in poorer countries) to choose a projection that works well for navigation even if the primary purpose of the map was not for navigation. If one can afford to prepare and print only one map it is prudent to create that map so it is well-suited for navigation given the likelihood that even maps prepared for other purposes might end up being pressed into service for navigation. Thus the Mercator has been widely used even in maps not primarily aimed at navigation. Indeed, many voyagers on a budget have used Mercator maps for navigation that were originally created for other purposes.

Far from promoting a thin band of First World latitudes, if anything, the Mercator grossly disfavors such latitudes by overstating the geographic importance of regions populated by Arctic peoples and high latitude indigenous populations. Perhaps what really annoys Peters advocates is that the Mercator overemphasizes the habitat of Third World populations in regions other than those favored by Peters partisans. Even so, to suggest that because the Mercator is bad one should employ the "Peters projection" doesn't hold water given the many projections other than the Mercator which are both readily available and clearly superior to the Gall should equal representation be the goal (which, of course, it is not for Peters partisans, who seek an unequal representation for their side).

From a purely technical perspective, falling for "Peters projection" propaganda does a disservice to Third World nations by applying a projection that denies them the technical benefits of more effective cartographic tools. It is a sad statement of the triumph of politically-motivated publicity that organizations such as UNESCO have adopted the "Peters projection" despite the poor cartographic properties of the projection. While some partisans apologize for the Peters movement by stating that "all maps are political," only someone bereft of technical knowledge could so claim given the many profoundly useful technical characteristics of various projections in matters that have nothing to do with politics. See, for example, the Guide to Selecting Map Projections topic.

While there will always be propagandists eager to employ GIS to deceive, all friends of GIS can fight such deception by patiently working to correct misunderstandings fostered by the popular media and other inexpert information channels. A fair representation of the Earth upon a flat surface is a noble goal, the original reason for projections to have been invented in the first place. It continues to be a goal worth fighting for even when the effort is attacked for political reasons.

Just as Mercator himself continued his work even after the attacks upon him for his own cartographic work, GIS advocates can help push back against political pressure by patiently explaining the technical properties of projections and by providing examples of how modern computer technology can show our world in a fair and neutral way for the benefit of all concerned.
Mercator

The famous conformal cylindrical projection. It is excellent for the limited purpose of marine navigation and very poor for thematic presentation. See the Transverse Mercator topic for the transverse aspect of this projection.

Scale

True along the Equator and along two chosen standard parallels equidistant from the Equator.

Increases with distance from the Equator to infinity at the poles.

Scale is constant along any given parallel; same scale at parallel of opposite sign (north +, south -) and is the same in all directions near any given point.

Distortion

Infinitesimally small circles of equal size on the globe appear as circles on the map (indicating conformality) but increase in size away from the Equator (indicating area distortion). Great distortion of area in polar regions. Conformality (and therefore local angle preservation) fails at the poles. Low distortion near the Equator.

Usage

Designed and recommended for navigational usage because of straight rhumb lines; standard for marine charts. Recommended and used for conformal mapping of regions predominantly bordering the Equator. The Mercator projection attained great fame because of its utility for marine navigation.

One unfortunate result of this fame is that it subsequently was frequently and inappropriately used as a thematic world map in atlases and for wall charts. It presents a highly misleading view of the world because of the excessive distortion of area away from the poles. The classic schoolroom map of the Earth in Mercator projection that has convinced generations of school children that Greenland is as large as North America is a case in point.

See the Gall projection topic for additional discussion of the Mercator projection, why it is especially useful in the Third World and the evil political attack on its usage.
Origin

Presented by Gerardus Mercator (1512-1594) of Flanders in 1569 on a large world map "for use in navigation".

Gerardus Mercator

A 1532 masters graduate of the University of Louvain, Mercator was originally known as Gerard de Cremere. He worked mainly in Louvain and Duisburg, where he produced most of his famous maps and was appointed Court Cosmographer to Duke Wilhelm of Cleve.

Mercator had moved to Duisburg in 1552 after the University of Louvain interceded to have him freed from prison where he had languished for seven months after being charged with heresy in 1544 after travels related to his cartographic work. He was the first to use the term atlas to refer to a collection of maps.

Mercator also invented a process for mass production of globes. In earlier times globes were produced using a labor-intensive process of engraving directly upon solid spheres. Mercator created papier-mâché, hollow globes to which he glued mass produced, hand-colored, paper engravings shaped in the now-familiar 12 gores (shapes narrowing towards the poles) plus two circular polar end caps. He produced several hundred terrestrial and celestial globes mounted in wooden stands, of which 22 matched pairs are still extant.

Over 450 years after his imprisonment, Mercator continues to be assaulted by the technologically inept but politically well-connected. See the discussion of the propaganda movement using the Gall projection in the Gall topic for the latest political assault on Mercator.

Limiting Forms

A Transverse Mercator projection is achieved when the Mercator projection's cylinder is rotated about the Earth so that instead of the Equator being the central ring of the cylinder a Meridian (that is, a longitude line) becomes the central ring.

Options

Specifying a non-Equatorial latitude origin causes an oblique projection.
Miller Cylindrical

A cylindrical projection that is neither conformal nor equal-area. A compromise between the Mercator and other cylindrical projection that attempts to eliminate some of the scale exaggeration of the Mercator. An "American version" of the Gall projection.

**Scale**

Only the Equator is true to scale.

**Distortion**

Only the Equator is free of distortion.

**Usage**

Used for world maps and in several atlases, including the National Atlas of the United States prepared by USGS in 1970.

**Origin**

Presented in 1942 by Osborn Maitland Miller of the American Geographical Society.
Modified Transverse Mercator
A USGS projection developed in 1972 for the revision of a 1954 map of Alaska. Consisting of an adaptation for the Universal Transverse Mercator projection series, it is not really a cylindrical projection but is better understood as an approximation to the Equidistant Conic projection. This projection is used for maps of Alaska and the Aleutian - Bering Sea region and is best left for experts only.
Oblique Mercator

An oblique cylindrical projection that is conformal but not equal area. The Oblique Mercator for the sphere is equivalent to a regular Mercator projection that has been altered by wrapping a cylinder around the sphere so that it touches the surface along the great circle path chosen for the central line instead of along the Earth's Equator.

Manifold provides four forms of the Oblique Mercator:

- Oblique Mercator (A)
- Oblique Mercator (A, centered)
- Oblique Mercator (B)
- Oblique Mercator (B, centered)

All four projections use the same formulae, but their initialization rules differ as noted below.

Scale

True along chosen central line, a great circle at an oblique angle or along two straight lines parallel to central line.

Distortion

Rapidly increasing distortion away from the great circle central line.

Usage

Used to map regions, such as the Alaska panhandle, that lie along oblique paths or great circles as opposed to regions that are North-South or East-West in extent. Used for grids on maps of the Alaska panhandle, for mapping in Switzerland, Madagascar and Borneo and for atlas maps of areas with greater extent in an oblique direction.

Normally, the Oblique Mercator is used only to show the region near the central line and for a relatively short portion of the central line.
Origin

Developed in 1900 - 1950 by Rosenmund, Laborde, Hotine and others.

Limiting Forms

The regular Mercator is the limiting form using the Equator as the central line, while the Transverse Mercator is the limiting form using a meridian as the central line.

Options

There are four variations of Oblique Mercator in Manifold: A and B variations as well as centered and non-centered versions. These differ from one another in how their options are used to specify the projection.

- A versions take a Center Latitude and two additional points defined by 1st and 2nd Latitude and Longitude.

- B versions take a central point defined by a Center Latitude and Center Longitude together with a Center Line Azimuth.

- Centered versions unproject 0,0 to the center of the projection, while non-centered versions unproject 0,0 to the intersection of the Equator and the Prime Meridian.

The Oblique Mercator (B, centered) version is conceptually easiest for many people. To use it, specify the Center Latitude and Center Longitude for the desired center of the map. Specify the Center Line Azimuth to be the bearing line that runs in the approximate line of the main features of the map.

For example, the illustration at the beginning of this topic was created using Oblique Mercator (B, centered) using Center Latitude 0 (the Equator), Center Longitude -90 and a Center Line Azimuth of -30. The central point defined by the Center Latitude and Center Longitude is approximately in the middle of the North American and South American landmasses near the Isthmus of Panama. An azimuth of -30 is approximately the angle made drawn through the center of masses of North and South America.

The graticule in this example was originally created in a latitude / longitude drawing for the region near North and South America. After projection into Oblique Mercator (B, centered) those portions of the graticule not reasonably near the central line of the projection were very distorted. They were selected and deleted by hand.
A conformal cylindrical projection: The transverse aspect of Mercator projection. Also known as Gauss Conformal (ellipsoidal form only), Gauss-Kruger (ellipsoidal form only) and Transverse Cylindrical Orthomorphic. Shown greatly zoomed in since profound distortion occurs outside the target region.

Limitations

The accuracy of Transverse Mercator projections quickly decreases from the central meridian. Therefore, it is strongly recommended to restrict the longitudinal extent of the projected region to +/- 10 degrees from the central meridian. [The US Army standard allows +/- 24 degrees from the central meridian].

This requirement is met within all State Plane zones that use Transverse Mercator projections.

Scale

True along the central meridian or along two straight lines on the map equidistant from and parallel to the central meridian. Scale is constant along any straight line on the map parallel to the central meridian. These lines are only approximately straight for the projection of the ellipsoid, and will be the case within Manifold when ellipsoidal Earth models (the standards) are used.

Scale increases with distance from the central meridian, and becomes infinite 90° from the central meridian.

Distortion

Infinitesimally small circles of equal size on the globe appear as circles on the map (indicating conformality) but increase in size away from the central meridian (indicating area distortion).
Usage

Many of the topographic and planimetric map quadrangles throughout the world at scales of 1:24,000 to 1:250,000. Basis for the Universal Transverse Mercator (UTM) grid and projection. Basis for the State Plane Coordinate System in U.S. States having predominantly north-south extent. Recommended for conformal mapping of regions having predominantly north-south extent.

Origin

Presented by Johann Heighrich Lambert (1728 - 1777) of Alsace in 1772. Formulas for ellipsoidal use developed by Carl Friedrich Gauss of Germany in 1822 and by L. Kruger of Germany, L.P. Lee of New Zealand, and others in the 20th Century.

Options

Specifying latitude origin and longitude origin centers the map projection.

Pseudocylindrical

Eckert IV

A pseudocylindrical, equal area projection.

Scale

True along latitudes 40°30’ North and South.

Scale is constant along any given latitude and the same for the latitude of opposite sign.

Distortion

Free of distortion only at latitudes 40°30’ North and South at the central meridian.

Usage

Thematic and other world maps in numerous atlases and textbooks and for sheet maps.

Origin
Presented by Max Eckert (1868-1938) of Germany in 1906.

**Options**

Specify the *center longitude* parameter to center the map projection.
Eckert VI

A pseudocylindrical, equal area projection.

Scale

True along latitudes 49°16' North and South.

Scale is constant along any given latitude and the same for the latitude of opposite sign.

Distortion

Free of distortion only at latitudes 49°16' North and South at the central meridian.

Usage


Origin

Presented by Max Eckert (1868-1938) of Germany in 1906.

Options

Specify the **center longitude** parameter to center the map projection.
Goode's Homolosine

An interrupted, pseudocylindrical, composite, equal area projection.

The **Clip Coordinates** checkbox **must** be checked when projecting maps into Goode's Homolosine.

**Scale**

True along every latitude between 40°44' North and South and along the central meridian within the same latitude range.

**Distortion**

Same as the Sinusoidal projection between latitudes 40°44' North and South and is the same as the Mollweide projection beyond this range. The projection is intended to be shown as an interrupted projection.

**Usage**

Whole-world maps, especially in Goode's Atlas published by Rand McNally. Because it is intended as an interrupted projection it is unsuitable for accurately showing land forms (Greenland and Antarctica) that bridge the interrupted space between the projection lobes.

**Origin**

Developed in 1923 by J. Paul Goode (1862 - 1932) of the University of Chicago as a merging of the Mollweide with the Sinusoidal.

**Options**

The illustration above shows a map window with two drawing layers. The upper layer shows countries. The lower layer contains a grid of lines created with **View - Graticule**. Horizontal graticule lines that fell in the "interrupted" zone between the projection lobes were manually selected (with **SHIFT**-select box) and then deleted. Greenland and Antarctica were selected and deleted from the upper layer.

**Tech Tip**

Interrupted projections are not continuous coordinate systems. They employ multiple conversion domains with blank space between the different lobes of the conversion domains. It is absolutely essential to check the **Clip Coordinates** box in the **Projection** dialog whenever using such projections. This box causes Manifold to clip each object so that it exists only within the allowed conversion domain lobes and does not extend or cross
through disallowed blank space. This is a highly computationally intensive process so the **Clip Coordinates** box is not checked by default, so that significant overhead is not imposed if it is unnecessary.

Dealing with the separate conversion domain lobes of an interrupted projection requires a manual approach to creating graticules, since the graticule lines normally extend through the blank space between lobes. Use the following procedure:

1. Create a latitude / longitude drawing.
2. Use the View - Graticule tool to create the desired graticule, using the option to **Create** the graticule as line objects.
3. Project the drawing into the desired interrupted projection, making sure to check **Clip Coordinates**.
4. Edit the graticule lines by selecting undesired lines and deleting them, or by adding lines.

For many uses the fastest method is to add lines. Let's consider an example of creating a graticule for use with Mollweide Interrupted. Suppose we begin with a latitude / longitude drawing in which a graticule was created with lines every 10 degrees from -180 to 180 longitude and from -70 to 70 latitude.

![Graticule Example](image)

After projection into Mollweide Interrupted with **Clip Coordinates** checked, the lines that appear on the edges of the conversion domain lobes will have been deleted.

![Projection Example](image)

We can add lines by clicking Snap To Lines and then using the Insert Line tool to add lines between the "dangling" parallels. This goes very rapidly with less than a minute required to complete the graticule.
The result will be a graticule grid with lines restored that were deleted by Clip Coordinates. This procedure was used to create the graticule seen in the interrupted projection illustration above.
**Guyou**

A conformal projection computed with elliptical integrals in the same family as the Peirce projection. Although it could be considered an azimuthal projection, it is listed here as a pseudocylindrical projection because it is often used for "horizontal" display of world regions as are cylindrical projections.

**Caution:** This is a "one-way" projection in terms of accuracy in that once data is projected into the Guyou coordinate system it cannot be re-projected to other systems without some loss of precision. The definition of the projection in common use does not include formulae for the inverse transformation, so projecting from Guyou to latitude / longitude uses iterative approximations and thus can lose precision.

**Scale**

Not true anywhere. Scale is especially false in the corners of the projection, where scale is elongated, and at the center, where scale is compressed.

**Distortion**

The projection is conformal.

**Usage**

Useful for showing equatorial regions. In its complete form (the entire Earth), may be tiled infinitely to create a display that allows any point on Earth to be viewed in a direct line from any other point.

**Limiting Forms**

The Guyou and Peirce projections are transverse cases of each other. The Guyou is the equatorial aspect (hence the classification as a pseudocylindrical projection within Manifold) and the Peirce is the polar aspect (hence the classification as an azimuthal projection within Manifold).
Options

Specify the center of the projection by setting the **center longitude**.

Origin

Presented by French mathematician Emile Guyou (1843 - 1915) in 1886.

Guyou was elected as a member of the French Academy of Sciences in 1894 and is memorialized by a street in Toulouse as well as by the Guyou Islands group and Guyou Bay in Antarctica. At the time a Captain in the French navy, Guyou prepared a report on the magnetic results of the Antarctic expedition in 1897-1899 under Gerlache as well as the 1903-1905 expedition under Charcot.
Hammer

A modified azimuthal, equal area projection. Listed here as a pseudocylindrical projection because it is often used as a pseudocylindrical projection to display the whole world at once.

**Scale**

Decreases along the central meridian and the Equator with distance from the center.

**Distortion**

Moderate. Less shearing action on the outer meridians near the poles than there is in true pseudocylindrical projections.

**Usage**

Whole-world maps, also in interrupted and condensed forms.

**Origin**

Presented by H.H. Ernst von Hammer (1858 - 1925) of Germany in 1892.

**Options**

Specify the `center longitude` parameter to center the map projection.
A pseudocylindrical, equal area projection. Also known as Homolographic projection, Homalographic projection, Babinet projection, and Elliptical projection.

The Mollweide projection is also provided in interrupted form as the Mollweide Interrupted in Manifold.

**Scale**

True along latitudes 40°44' North and South.

Scale is constant along any given latitude and the same for the latitude of opposite sign.

**Distortion**

Free of distortion only at latitudes 40°44' North and South on the central meridian. Distortion is severe near outer meridians at high latitudes.

**Usage**

Occasional world maps, especially thematic maps. Combined with Sinusoidal projection to develop other projections such as the Goode Homolosine and the Briggs.

**Origin**

Presented by Carl B. Mollweide (1774 - 1825) of Germany in 1805. Also known as the Hammer-Aitoff. Mistakenly called the Aitoff projection in the early 20th century. The Briesemeister is a modified oblique Hammer.

**Options**

Specify the *center longitude* parameter to center the map projection.
Mollweide Interrupted

A pseudocylindrical, equal area projection. Also known as Homolographic projection, Homalographic projection, Babinet projection, and Elliptical projection. The Molleweide Interrupted projection is a variation of the Mollweide projection.

The **Clip Coordinates** checkbox **must** be checked when projecting maps into Mollweide Interrupted.

**Scale**

True along latitudes 40°44' North and South.

Scale is constant along any given latitude and the same for the latitude of opposite sign.

**Distortion**

Free of distortion only at latitudes 40°44' North and South on the central meridian. Distortion is severe near outer meridians at high latitudes.

**Usage**

The Mollweide Interrupted is used to display the oceans of the world. For example, for voyagers it's a nearly perfect projection to show the path of a circumnavigation via the Panama Canal and the Red Sea.

**Origin**

Presented by Carl B. Mollweide (1774 - 1825) of Germany in 1805.

**Options**

**Clip Coordinates** must be checked.

**Tech Tip**

Interrupted projections are not continuous coordinate systems. They employ multiple conversion domains with blank space between the different lobes of the conversion domains. It is absolutely essential to check the **Clip Coordinates** box in the **Projection** dialog whenever using such projections. This box causes Manifold to clip each object so that it exists only within the allowed conversion domain lobes and does not extend or cross through disallowed blank space. This is a highly computationally intensive process so the
Clip Coordinates box is not checked by default, so that significant overhead is not imposed if it is unnecessary.

Dealing with the separate conversion domain lobes of an interrupted projection requires a manual approach to creating graticules, since the graticule lines normally extend through the blank space between lobes.

Use the following procedure:

1. Create a latitude / longitude drawing.
2. Use the View - Graticule tool to create the desired graticule, using the option to Create the graticule as line objects.
3. Project the drawing into the desired interrupted projection, making sure to check Clip Coordinates.
4. Edit the graticule lines by selecting undesired lines and deleting them, or by adding lines.

For many uses the fastest method is to add lines. Suppose we begin with a latitude / longitude drawing in which a graticule was created with lines every 10 degrees from -180 to 180 longitude and from -70 to 70 latitude.

After projection into Mollweide Interrupted with Clip Coordinates checked, the lines that appear on the edges of the conversion domain lobes will have been deleted.

We can add lines by clicking Snap To Lines and then using the Insert Line tool to add lines between the "dangling" parallels. This goes very rapidly with less than a minute required to complete the graticule.
The result will be a graticule grid with lines restored that were deleted by **Clip Coordinates**. This procedure was used to create the graticule seen in the interrupted projection illustration above.
Robinson

A Pseudocylindrical projection that preserves neither scale nor area, but which presents an aesthetically pleasing view of the entire world.

The Robinson projection is unlike most other projections in that it is not constructed by a mathematical formula used to transform coordinates systems. It is instead constructed by reference to a table of transformation parameters for meridians and parallels with an interpolation for locations between those given in the table. The table was created empirically to achieve a pleasing effect when the entire world is displayed.

Scale

Not true anywhere.

Distortion

Not free of distortion anywhere but the severe distortion near outer meridians at high latitudes seems less objectionable than in other pseudocylindrical projections.

Usage

Overview maps of the entire world intended to present thematic data and not intended to be used for distance or area measurement. The Robinson projection is often recommended as a “good compromise” projection for world thematic maps.

Origin

Created by Arthur Robinson at the request of a commercial atlas publisher in the 1960’s. Computer assisted cartography played an essential role in the trial-and-error development of the table of transformation parameters and so represents an early use of computers in the evolution of cartography. Although this is a very popular projection for world thematic maps, it is not defined either in Snyder’s famous USGS books on map projections or in Robinson’s own “Elements of Cartography.” The only technical definitions for it occur in two obscure papers: Robinson’s original announcement of it and Snyder’s publication of a small paper in a briefly published, now defunct journal reporting a computational algorithm for implementing the Robinson projection. Manifold’s Robinson projection is based on the Snyder algorithm.

Options

Specify the center longitude to center the map projection. This is normally the 0 meridian.
Special Credits

The manifold.net team would like to thank Tau Rho Alpha of the Menlo Park USGS staff for his assistance in helping us gain access to the algorithms used in the Robinson projection. A representative of our team voyaged to Menlo Park and was distressed to find the only copies of Snyder's paper were located at a different USGS facility. Mr. Alpha generously took a stranger under his wing and helped us track down the data we needed. This is but one example of a life of generous outreach to help young and old, expert and novice alike to gain greater awareness of cartography. Tau Rho Alpha has not only helped experts learn to use projections better, he is the prime mover behind an effort to help school children learn geography by creating their own globes from paper cutouts.
Sinusoidal

A Pseudocylindrical, equal area projection with equally spaced parallels. Also known as Sanson-Flamsteed and Mercator Equal-Area.

Scale

True along every parallel and along the central meridian.

Distortion

Free of distortion along the Equator and along the central meridian. Severe distortion near outer meridians at high latitudes.

Usage

Atlas maps of South America and Africa. Good for countries with great north-south extent. Occasionally used for world maps. Formerly used for other continental maps and star maps. Combined with Mollweide projection to develop other projections such as the Homolosine and the Boggs.

Origin

Developed in the 16th century. Used by J. Cossin in 1570 and by J. Hondius in Mercator atlases of the early 17th century. Often called Sanson-Flamsteed projection after later users. Oldest current pseudocylindrical projection.

Options

Specify the center longitude to center the map projection.
Van der Grinten

A curved modification of the Mercator projection that is neither equal-area nor conformal. Although it is usually classed with the various pseudocylindrical projections it is not really a pseudocylindrical projection.

Scale

Only the Equator is true to scale.

Distortion

The central meridian and Equator are straight lines. All other meridians and parallels are arcs of circles. Great distortion in polar regions. Most maps using this projection do not extend past Greenland and the outer rim of Antarctica.

Usage

Used for world maps. Used by the National Geographic Society for their standard world map until 1988, which resulted in adoption of this projection by many other groups.

Origin

Presented in 1904 by Alphons J. van der Grinten of Chicago writing in a German geographical journal and patented in the United States in 1904. Van der Grinten originally invented two projections, the first of which is known as the "Van der Grinten I" or simply the "Van der Grinten" and the second of which is (confusingly) known as the "Van der Grinten IV." See the Van der Grinten IV topic for the Manifold implementation of that second projection.

Van der Grinten's second projection is known as "IV" because after van der Grinten's original publication Alois Bludau in 1912 presented a variation of the first van der Grinten projection that became known as the "Van der Grinten II," as well as a second variation in 1912 based upon the second of van der Grinten's original projections and given the name of "Van der Grinten III."

Thus the original two projections are known as Van der Grinten I and IV, while Bludau's two variations are known as II (based upon the first original) and III (based upon the second original).

The "Van der Grinten" projection used within Manifold is the first of the series and was invented in 1898. It is the best known as the result of wide use by the National Geographic Society and the one commonly used. Van der Grinten published the projection using a
geometric construction. Manifold uses the 1979 formulae published by Snyder.

**Limiting Forms**

Used only in the spherical form.
Van der Grinten IV

A curved modification of the Mercator projection that is neither equal-area nor conformal. Although it is usually classed with the various pseudocylindrical projections it is not really a pseudocylindrical projection.

Scale

Only the Equator is true to scale.

Distortion

The central meridian and Equator are straight lines. All other meridians and parallels are arcs of circles. Great distortion in polar regions. Most maps using this projection do not extend past Greenland and the outer rim of Antarctica.

Usage

Used for world maps.

Origin

Presented in 1904 by Alphons J. van der Grinten of Chicago writing in a German geographical journal and patented in the United States in 1904. Van der Grinten originally invented two projections, the first of which is known as the "Van der Grinten I" or simply the "Van der Grinten" and the second of which is (confusingly) known as the "Van der Grinten IV." See the Van der Grinten topic for the Manifold implementation of the first projection.

Van der Grinten's second projection is known as "IV" because after van der Grinten's original publication Alois Bludau in 1912 presented a variation of the first van der Grinten projection that became known as the "Van der Grinten II," as well as a second variation in 1912 based upon the second of van der Grinten's original projections and given the name of "Van der Grinten III."

Thus the original two projections are known as Van der Grinten I and IV, while Bludau's two variations are known as II (based upon the first original) and III (based upon the second original).

Limiting Forms

Used only in the spherical form.
Projections
Wagner IV

A pseudocylindrical, equal-area projection invented by Putnins and called the Putnins P2" but almost universally known today in GIS circles as the "Wagner IV".

Scale

True along latitudes 42°59' North and South. Constant along any given latitude and the same for the latitude of opposite sign.

Distortion

Distortion is not as extreme near outer meridians at high latitudes as it is on pointed-polar pseudocylindrical projections, but there is considerable distortion throughout polar regions. Free of distortion only at latitudes 42°59' North and South at the central meridian.

Usage

Thematic world maps.

Origin

Presented in 1934 by Reinholds V. Putnins of Latvia.

An identical projection, the Wagner IV projection, was presented by Karlheinz Wagner of Germany in 1949.
Wagner VII

A modified azimuthal, equal-area projection that was the seventh in a series of new projections presented by Wagner in his cartographic work.

**Scale**

Decreases along the central meridian and the Equator with distance from the center of the projection.

**Distortion**

Considerable shape distortion in polar areas.

**Usage**

Thematic world maps, such as climatic maps prepared by the U.S. Department of Commerce.

**Origin**

Presented by Karlheinz Wagner of Germany in 1941, the Wagner VII is a modification of the Hammer projection and is also known as the Hammer-Wagner projection.
Winkel Tripel

A modified azimuthal projection that is neither conformal nor equal area. This is a low precision projection as normally used since it is calculated for a sphere and not an ellipsoid. However, when presenting whole world maps the difference between a spherical and ellipsoidal Earth is invisible.

Scale

True along the central meridian. Constant along the Equator.

Distortion

Moderate except near outer meridians in polar regions.

Usage

Whole world maps. After languishing for many years, the Winkel Tripel is becoming fashionable once more as a result of usage by the National Geographic Society.

Origin

Presented by Oswald Winkel (1873-1953) of Germany in 1921. The Winkel Tripel is obtained by averaging coordinates of Cylindrical Equidistant and the Aitoff projections. Winkel coined the name "tripel" (meaning "triple" in English) to reflect the three step process: project into Cylindrical Equidistant, project into Aitoff and then average the two projections.

Satellite View

Space Oblique Mercator

A modified cylindrical projection with the map surface defined by a satellite orbit. Basically conformal, especially in region of satellite scanning.

Scale

Continuously true to scale along ground track of satellite.

Distortion
Near-zero distortion along ground track of satellite.

Usage

Used for mapping images from Landsat series satellites. May be used for any satellite orbiting the Earth in a circular or elliptical orbit and at any inclination.

Origin

Developed in 1973 - 1979 by Alden P. Colvocoresses, John Parr Snyder and John L. Junkins. Colvocoresses was the first to realize that such a projection was needed and mathematically feasible, and in 1974 defined it geometrically. Snyder created the definitive formulas for the projection in 1977 - 1979, while less complete formulas were also created by Junkins in 1977. Snyder provided a complete derivation for orbits of any ellipticity in 1981.

Optional Forms

The Space Oblique Mercator for Landsat is a pre-computed form of the Space Oblique Mercator projection that allows specification of the projection by citing a Landsat Orbit Number and Orbit Path Number.

Historical Note

J.P. Snyder is revered throughout professional cartography as one of the giants of computational cartography. He arrived late in life to his career with USGS after a previous career as a chemical engineer. The Space Oblique Mercator projection formulae created by Snyder launched his second career in cartography.

USGS had been unable to develop projection formulae that implemented Colvocoresses's geometric definition and made an open appeal for help at the 1976 geodetic conference at Ohio State University. Snyder attended the conference while on vacation (he had a hobby interest in cartography and would schedule his vacations to allow him to attend cartographic conferences) and heard the appeal. He developed the required formulae on a pocket calculator and donated them to USGS. In 1978 he joined the professional staff of USGS and began his second career at the age of 52 as a computational cartographer.

Universal Polar Stereographic

Universal Polar Stereographic

An azimuthal projection that is conformal. A standard variation of the Stereographic projection in polar aspect that is used with Universal Transverse Mercator (UTM) projection systems to represent polar regions.

Scale

True only where the central latitude crosses the central meridian or, alternatively, along a circle concentric about the projection center (or a parallel on the polar aspect). Scale is constant along any circle having its center at the projection center, but scale increases moderately with distance from the center within a hemisphere.

Distortion
Only the center or the circle of true scale (if not the center) is free from all distortion. Areas grow greater the farther from the center, albeit in a conformal manner.

**Usage**

Employed in the UTM system to show North and South polar regions. Used from the North Pole to latitude 84 degrees North and from the South Pole to latitude 80 degrees South. In Universal Polar Stereographic projections the scale is reduced to 0.994, resulting in a standard parallel of about 81 degrees 07 minutes North or South.

**Limitations**

The Universal Polar Stereographic projection is used only from the North Pole to 84 degrees latitude North and from the South Pole to 80 degrees latitude South.

**Origin**

Apparently developed in polar aspect by Egyptians and Greeks by the 2nd Century BC. Hipparchus was apparently the first Greek to use it and is generally considered its inventor.

**Universal Transverse Mercator**

**Universal Transverse Mercator (UTM)**

The UTM system applies the Transverse Mercator projection to mapping the world, using 60 pre-defined standard zones to supply parameters. UTM zones are six degrees wide. Each zone exists in a North and South variant.

Europeans can best understand UTM by thinking of it as a world-wide version of the Gauss Kruger system, which is also based on a regular system of Transverse Mercator projections that each map a zone six degrees wide. (Americans can think of the Gauss Kruger system as a version of UTM with zones defined for European and Asian coverage.)

- **Universal Transverse Mercator**
  - The Northern Hemisphere projections for the infamous UTM system consisting of 120 zones (60 different zones with North and South variants of each). Originally developed for military use and now widely misused in civil mapping.

- **Universal Transverse Mercator (South)**
  - The Southern Hemisphere projections for UTM. These are mainly distinguished by each having a Northing parameter of 10 million so that no coordinates need involve negative numbers.

To use a UTM projection, click on the **Universal Transverse Mercator** heading in the projections categories for Northern Hemisphere regions and on the **Universal Transverse Mercator (South)** heading for Southern Hemisphere regions. Choose the UTM zone desired.

See the Projections Tutorial topic for more info on projections.

**Limitations**
The accuracy of any Transverse Mercator projection quickly decreases from the central meridian. Therefore, it is strongly recommended to restrict the longitudinal extent of the projected region when using Universal Transverse Mercator projections to ±6 degrees from the central meridian.

This requirement is met within all State Plane zones that use Transverse Mercator projections.

**Each UTM Zone is a Different Projection**

The Mercator projection maps the world onto a cylinder where the central ring of tangency is the Earth's Equator.

Near the Equator, the Mercator projection provides low distortion. Away from the Equator distortion becomes very high. This limits the utility of the Mercator projection to regions near the Equator. That is a big limitation because most places that people live (and thus, most of the regions that people most frequently map) are located not along the Equator but along North-South directions, such as from North America to South America.

Turning the Mercator projection's cylinder so that it is tangent to the Earth along a meridian (longitude line) instead of the Equator results in what is called a **Transverse Mercator** projection. If we created a Transverse Mercator projection that had a meridian as the central ring of the cylinder we could make local maps anywhere along the North-South line of tangency. If the maps are limited to the thin, vertical region near the meridian of tangency they will be relatively free of distortion.

The problem is that any Transverse Mercator projection created by choosing any one meridian as a line of tangency is useful only near that meridian. If we pick a North-South line running through Athens we can make maps all the way from Scandinavia down the length of Africa, but any maps using this projection in North and South America would be hopelessly distorted.
The **Universal Transverse Mercator** system of projections deals with this by defining 60 different standard projections, each one of which is a different Transverse Mercator projection that is slightly rotated to use a different meridian as the central line of tangency. Each different centerline defines a **UTM Zone**. The "UTM Zone" is a shorthand way of naming a specific, different projection that consists of a Transverse Mercator projection using a different meridian as the centerline. By rotating the cylinder in 60 steps (six degrees per step) UTM assures that all spots on the Earth will be within 3 degrees of the centerline of one of the 60 cylindrical projections.

To map any spot on Earth, one picks the UTM Zone centerline that is closest to it and then makes a map using that "UTM Zone" cylindrical projection.

**UTM Zones should not be Combined**

Novice UTM users usually do not realize that each UTM Zone is in fact a different projection using a different system of coordinates. New users of UTM therefore will frequently attempt to "combine" different maps created in different UTM zones into one map with the expectation that the combined map will show all objects with low distortion as did the original maps. The motivating factor is often a desire to create a map centered on a region of interest that spans several UTM zones or which is centered between two zones. Such plans fail to take into account that UTM is an intrinsically inflexible system. In effect, the UTM system assumes objects from different zones will never be seen together in the same map.

Combining objects from different UTM zones into a map that is projected using only one of those UTM zones will result in distortion in the locations and shapes of the objects that originated in a different zone map. Geographic shapes that look good in a transverse Mercator projection centered upon a given UTM zone line will be very distorted when illustrated in a UTM projection centered upon a different zone line.
The illustration above shows part of Europe projected into UTM Zone 2 in the yellow map. Overlaid on the yellow map is an Orthographic projection centered on the same map center shown in blue color. The numbers are positioned at the center of UTM Zones 1, 2 and 3.

The Orthographic map is essentially accurate over the entire illustration. In contrast, the UTM map is highly inaccurate only one half zone away from the "home" zone. Note that it distorts the coast of France so much that it has France (in the yellow, UTM projection color) crossing the Channel.

If we need to combine objects from several different UTM zones, the correct solution is to choose a different projection (such as a conic or azimuthal projection) for the combined map that provides low distortion over the entire region of interest. The illustration above shows a Lambert Conformal Conic projection in black outline and darker blue color overlaid over the Orthographic projection. Note that both projections are so close to each other it is difficult to pick out places where they differ. For example, in the region of France where the UTM projection had the continental landmass crossing the channel there is a very slight North/South offset but otherwise the two projections are virtually the same.

Remember, although no projection is perfect for all uses some projections are better than others in the uses for which they were designed. UTM was designed to map objects within one zone at a time. It is a very bad choice if objects from several zones must be shown together on the same map.
Comments

Like the State Plane Coordinate System or Gauss Kruger, UTM is a living fossil. It was created for use in expert hands as a means of dealing with the technological limitations of an earlier era. When used with skill as originally intended it still functions well within its intended uses. The problem with UTM (as with the State Plane and Gauss Kruger systems) is that it is constantly misused in civil applications by inexpert users who do not realize the limitations built into the system.
Networks

Network User's Introduction

Manifold is often used to analyze transportation networks. In this usage Manifold is very different than most other GIS or mapping products because Manifold aims at more mathematical-oriented networking analysis. Most other GIS and mapping products use bits and pieces of network math to solve highly specific tasks. This makes them good for specific tasks that are well known and accepted but less useful for extending the art.

The Manifold approach is to embed within the main system a general set of capabilities for analyzing networks using the basic building blocks of network mathematics. For specific tasks involving transportation analysis, road networks and other "traditional" GIS tasks manifold.net publishes optional add-on products in the Business Tools package.

It should be emphasized that the basic system can do everything done in the add-in products. Manifold has an unusually rich array of network analytic tools that experts can use in highly sophisticated ways to accomplish almost any network analytic task.

The mathematics of network analysis is graph theory and the same graph theoretical algorithms are used whether one analyzes paths through computer networks or paths through road networks. When Manifold was created it seemed a shame to limit such a fine analytical instrument to only analyzing some networks when it could be applied to all networks. Therefore, Manifold was built as a general tool that can be used for networks of all kinds.

This approach is not without its cultural conflicts, since GIS users who do networking analysis on transportation networks do not share the same viewpoints as computer people who analyze computer networks. Neither of these two groups shares the same culture as mathematicians who develop and extend graph theory. We have tried to resolve such social issues within the documentation by making references that are hopefully understood by all groups.
About Networks

A network consists of nodes that may or may not be connected together by links. Networks are often illustrated with dots or circles representing the nodes and straight or curved lines between nodes representing the links. Nodes may be joined by more than one link. There may even be nodes that are not connected by any link. A network drawn in a GIS or CAD system will be drawn using points for the nodes and lines for the links.

The most important thing about a network for new users to understand is that it is defined by its nodes, and not by the links. People new to network analysis will often look at a drawing of lines and think "Aha! That's a network!" However, if there are no points in that drawing there is no network in the drawing. Since GIS users will often wish to use a system of lines as a network even if it contains no nodes, Manifold takes a less strict view: default operators in Manifold will treat as a network any system of lines where the endpoints of the lines exactly coincide even if no points exist at the ends of the lines.

We could continue calling these objects "points" and "lines"; however, the terminology normally shifts into the use of the words "nodes" and "links" when a GIS map of points and lines is being treated as a network. That's because (as we will see below) there are certain implications when one says a line is a link in a network. For example, a link always runs between two nodes. There is no such thing in a network as a "line" all by itself without a point at either end.

Although networks are used to represent real systems which actually look like networks, for example, a road system, networks may also be used to analyze physical systems which don't look at all in real life like dots connected by lines. For example, networks may be used to represent the relationships between elements in complex systems such as the assembly of complex machines, scheduling tasks, or the operation of compilers.

Network theory has handy names to describe key qualities of networks, links, and nodes. Most of the time these names are quite clear and intuitively obvious. For example, a link which joins a node to itself is called a self-loop.

Links may be directed or undirected. A directed link is like a one-way street: it comes out of one node and goes into another node. Directed links are also called arcs. Networks which include directed links are called directed networks or oriented networks. Networks that have only undirected links are called undirected networks or unoriented networks. Directionality is implied in links drawing in GIS data by the sequence of coordinates that define a line.

Transportation networks involving a mix of one-way and two-way streets are often modeled as directed networks consisting of nodes joined by one or two directed links. Two-way streets are modeled as two directed links, one going each way, between nodes. However, directed links are only useful to the extent that network functions take such directionality into account. Most default Manifold network operators will ignore directionality.

Links and nodes in networks may also have weights associated with them. Weights are simply values in some data field associated with the link or node that represent some parameter that is important to the task at hand. For example, a network that represents a communications network might have a weight assigned to each link where the weight is a number giving the flow capacity of that communications link.

To take another example, consider a network that represents an interstate road system with a node for each city or major highway intersection with links between each node representing the interstate highway links between each city or intersection. Such a network might have several weights on each link representing values such as the measured physical length of that link, the maximum speed limit, the number of lanes, the average travel time over the link and so on. Nodes might have weights representing the total population in the city, average income, how many dealers we have in that city, what the average sales per dealer are, and so forth.

For the mathematically inclined...
When we talk about "networks" and "network theory" we are really discussing a branch of mathematics called graph theory. Although most people use the word "graph" in business to mean a diagram such as a bar chart, in mathematics the word "graph" refers to a mathematical object that consists of nodes that may or may not be joined by links.

In casual use, graphs are also called networks, for example as in "a road network," even though that’s not always strictly true. Since Manifold is sold mostly to non-mathematicians we use a more relaxed commercial terminology and use the term network throughout.

**Historical Note**

Leonhard Euler

Graph theory as we known it in modern times was founded by Leonhard Euler, born 15 April 1701 in Basel, Switzerland, and died 18 September 1783 in St. Petersburg, Russia. Perhaps the most prolific mathematician of all time, Euler founded graph theory while writing a discourse on the Bridges of Konigsberg problem.

The citizens of this Russian city enjoyed strolling the city while crossing the seven bridges. A popular entertainment was trying to figure out if one could walk over all seven bridges without repeating a bridge. Euler proved this was not possible.

Euler joined the St Petersburg Academy of Science in 1727 and served there until 1741 as a both a professor of physics and mathematics. He lived in Berlin for 25 years from 1741 until returning to Russia in 1766. Although almost entirely blind in the years after 1766, from then until his death in 1783 he produced almost half of his prodigious life’s product of over 880 books and papers. He was able to continue work by relying on his extraordinary memory.

**Pronunciation**

In English, "Euler" is pronounced as "oil - er" with the stress on the first syllable.
**Drawing Networks**

For most networks, usually what is most important is the relationship between the nodes and links, and not how they are drawn. That's why we can usually morph drawings of abstract networks to make them more comprehensible without losing key information. Most geographic networks should not be altered since their physical shape is often exploited to calculate values such as the length of links that we might wish to use in network analysis.

Small networks usually are easy to illustrate with simple drawings. Nodes are represented as points or small circles and links are represented as straight or curved line segments between the nodes. Directed links are supplied with arrows indicating the direction. Links in such a drawing can intersect with each other. By convention, cross points are not considered to be nodes of a network unless a node (i.e., dot or small circle) is drawn at the intersection.

Although one may illustrate a network with a drawing, note that it is the network (with its relationships between nodes and links) which is the mathematical object and not the picture drawn of it: different "pictures" may be drawn to represent the same network. For example, the following two figures illustrate the same network even though the pictures are different. Can you see why?

![Network Diagrams](image)

Not just any network can be drawn in a two-dimensional plane without the intersection of links. For example, the following illustration shows a network that cannot be drawn in two dimensions without intersections. As with the previous example, the same network has been drawn two different ways.

![Non-planar Network](image)

A network that can be drawn in two dimensions without intersections is said to be **planar**. For some problems the possibility of drawing a network on a plane without intersections is not essential; for other problems it is very important. The network illustrated is a **non-planar** network, and also happens to have the smallest number of nodes with which it is possible to draw a non-planar network.

In the "pure" mathematics of networks, as we have seen above, the same network can be drawn in different figures that except for the relationships they preserve between the nodes and links do not look at all alike. Manifold can be used in "pure" network theory to analyze networks in abstract coordinates as well as in applied network theory to analyze subjects like road networks using network concepts. In the case of computer networks sometimes the actual shape of the network is important, as in the case of a specific wiring layout for an office.
LAN, and sometimes only the abstract shape of the network is important, such as a map of links in cyberspace. At times both the physical shape and the abstract shape of a network are important.

In geographic applications we will use maps of transportation networks or other drawings where the shape of the figures is extremely important, because the shape of the figures contains important information. So, for example, in the case of a road network the shape of the roads conveys important information about their location, length, and so on. If we morph a drawing of a road network we may preserve the topological relationship between highway intersections and such but we will lose key data such as the geometric ability to calculate the length of a road by measuring the length of the link.

Let’s take for example a road system where interstate highways are links and cities or intersections between major highways are nodes. There are very many useful analyses that can be done on such a transportation network without ever having to render it as anything more than an abstract collection of lines and points. In fact, many useful analyses can be done simply from matrices listing nodes, links, and their weights without doing any rendering at all. However, even more can be done using the additional data that comes from examining the network as a real geographic map.

For example, suppose we have a real map of the network showing all the twists and turns of the road system that occur in real life but we don’t have the actual distances between each link. We can apply computational geometry to the map and compute what the distances are and then save each distance as a weight on each link. Manifold is frequently used for such tasks.

At times networks are used together with computational geometry, for example, in drawing Gabriel networks or relative neighborhood networks, where the entire essence of the resultant network derives from how it is drawn. In such cases, the precise physical way in which the network is drawn matters very much. In fact, in such networks the geometric proximity of one node to another is one of the factors used to determine if the two should be joined with a link.

When analyzing physical systems with networks it is important to strike a healthy balance between using networks that are drawn to look like the systems they model and more abstract renderings that can reveal relationships that the ordinary physical layout conceals.
Not All Maps are Networks

A typical use of Manifold System is to load an electronic map showing a road network and to then do some network analysis on the road network. Since points connected by lines are automatically treatable within Manifold as nodes and links, usually this works just fine. However, one must be careful to note that not all data sets are immediately usable as networks. Some data sets may require processing with Manifold commands to fix problems before they may be used in network analysis.

Some Background

Recall that a network is a set of nodes that may or may not be connected by links. Although single nodes unattached to any links are OK in a network, there is no such thing in network theory as a link without a node at each end, nor is it OK for one link to "connect" to another link without a node in between.

In Manifold, the elementary geometric objects are either Points, Lines, or Areas. Geometrically, Lines are really polylines. All lines, even what appear to be smoothly curved lines, are actually seen at high magnification to be made up of straight-line segments between the coordinates that define the line. Lines are simply lists x y coordinates that define the joint between the straight portions which together make up a line. A coordinate location that defines a line is not a point or treatable as a node, it is a purely internal item that is used to draw the line. Note that even if many straight portions between many coordinates are involved, the geometric Line object they define is still a single object in Manifold.

Lines are not Links without Points at Each End

Points are automatically treated as nodes and Lines between Points as links in network work within Manifold. If a Point happens to be positioned exactly at the same location as one of the end coordinates of a Line, then for network theory purposes that Point as a node is attached to the link that line represents. If the Point is positioned anywhere else other than exactly coincident with the end of a line it is an unattached node.

Note that as a matter of geometry it is perfectly possible for a Point to be drawn somewhere in the middle of a Line. However, in a network theory context, that Point would be regarded as an unattached node even though the drawing illustrated in a map window seems to show the Line running right through that Point. Remember, if a Point is not exactly at the end of a line it is not attached as a node to that link.

Lines that do not have Points at each of their ends are not links. From a network perspective, they do not exist.

It will be immediately obvious to the experienced CAD or GIS user that the above requirements for networks will encounter all sorts of problems with real life maps of lines and points that one may wish to use as networks. A typical problem is that a user gets handed a "road map" drawn in AutoCAD .dxf where all the roads are illustrated as lines. There are no points at all in the map, just a bunch of lines the ends of which may or may not be coincident when viewed at high magnification.

Since GIS users will often wish to use a system of lines as a network even if it contains no nodes, Manifold takes a less strict view: default operators in Manifold will treat as a network any system of lines where the endpoints of the lines exactly coincide even if no points exist at the ends of the lines.

Note: To add points (nodes) to the ends of all lines, use the Node Points transform operator using the Transform toolbar. It’s not a bad idea to have true networks in case user-supplied scripts or more rigorous operations insist that the map in use contain a true network.

Problems with Data Sets

There are several classes of common errors that might affect the use of a given data set as a network.

Points might not be coincident with the ends of Lines. The Point is an unattached node and as far as any network function (like routing) is concerned, the line does not exist in the network.
One Line may end at a given location and another Line might start at that location but there may be no Point at that location. The two points are not joined by a link, since the Lines between them do not exist as far as the network is concerned. Or, in a variation on this theme,

What appears to be a single Line (link) between nodes is actually made up of several Lines without nodes at their coincident ends. These nodes therefore are not joined by a link. Even though the map’s line objects abut each other, they are separate line objects. If there’s no Point at the exact ends of each of those lines they are not links and so do not exist in a network sense.

A Point may be coincident with one end of a Line, but there may be no Point at the other end. This is still an unattached node, since the "stub" line does not exist as a link.

A Line may have a Point at each end, but the next Line in what is intended to be an adjacent link may not have its end exactly coincident with the previous Line and Point. If interpreted as a network, the illustration shows a path consisting of three nodes joined by two links, plus a fourth, unattached node.

The ends of Lines intending to be incident to the same Point may not all be coincident, not with each other or with the Point. Taken as a network, this illustration shows three unattached nodes and no links.

All of the above problems often occur when software designed for freehand drawing (but which is utterly ignorant about networks) is used to create drawings intended for use as networks. The above problems can also occur when network-aware software such as Manifold is used in a sloppy fashion to draw new maps. For example, if we fail to use Snap correctly when drawing new lines it is easy to create lines the ends of which are not exactly coincident with points.

It’s easy to think that the Points in a given layer are connected by Lines when closer magnification will reveal they are not actually connected by the Lines. Perhaps the points lie near to the ends of the Lines, but they do not exactly coincide, or the Points are positioned on Lines at some middle area between the ends. It’s easy to think that just because there is a kink or sharp angle in a line that the kink marks the joint between two Lines and thus putting a Point there will attach it. However, such a kink might not mark the spot where two Lines come together, it just may be a sharp change of direccio in a single Line.

CAD drawings often have this problem, since many CAD renderings are drawn to a level of accuracy which makes lines and points “look connected” at the scales at which the drawings are normally viewed or printed. However, very frequently at high magnification the ends of the lines don’t coincide, the points are slightly detached from the lines and so on. Even a microscopic break is a gap for an automated command. One way of remediying this is to use Snap when drawing lines and points, so that the cursor snaps to exact locations.

Tech Tip:

Run the Normalize Topology transform toolbar operator to attach points and lines that are not attached within the network.

See Also

Network User's Introduction
Internet Map Server

Map Server Overview

The Manifold Internet Map Server (IMS) is an integral part of all Manifold editions except Personal Edition. When Manifold System Professional Edition or above is installed on a Windows web server, that machine will acquire the ability to host IMS web sites. Manifold IMS provides the following capabilities:

- **An Internet Map Server** for GIS-enabled websites using standardized templates that allow people visiting with ordinary browsers, such as Internet Explorer (IE), to view and query maps and other components within Manifold projects.

- **A programmatic interface** to the Manifold Internet Map Server allowing creation of GIS-enabled websites that provide access to Manifold projects, such as viewing, querying and a wide variety of operations with data in Manifold projects.

- **An administration page** for administrative access to IMS through the web.

- **Creation of a Manifold Image Server** interface page, allowing remote applications and Manifold System clients to fetch image tiles on demand for served images, much as how TerraServer or other image servers provide access to images.

- **OGC WFS server** functionality to enable OGC WFS clients to interact with vector data in the Manifold project, including altering data within served components.

- **OGC WMS server** functionality providing image serving to OGC WMS clients, including to Manifold System clients fetching linked images through OGC WMS.

Note that Manifold System is both a client and a server for OGC WMS and the Manifold Image Server interface; however, although Manifold IMS can be a server for OGC WFS, Manifold does not yet provide OGC WFS client capabilities. Manifold provides OGC WFS server capabilities so that those users who are using OGC WFS clients can get data from a Manifold IMS website.

Web Requirements

This topic assumes the reader is familiar with Microsoft Internet Information Server (IIS) as well as web programming using HTML and Active Server Pages (ASP or ASP.NET). Creating web sites with IIS is a straightforward process supported by a vast array of books and other educational resources. If you are not familiar with the operation of IIS, please get a good book on the subject and learn elementary IIS operation and administration before attempting to work with Manifold IMS. The IMS topics in this documentation provide some tips on working with IIS and Windows, but they are not a replacement for reading and understanding Microsoft documentation on IIS and Windows.

It is not at all difficult to create and publish a simple web site with Microsoft IIS; however, if you are unable to do this you also will not be able to publish a web site that uses Manifold IMS: please resolve any difficulty with your IIS installation before beginning work with Manifold IMS.

Before attempting to use Manifold IMS, please take a moment to create an ordinary .HTML web site on your server and then browse it from a different machine. If you cannot do this, stop. Do not attempt to work with Manifold IMS until you can get a simple web site up and running. This step is recommended even for experts: it is amazing how often people are "sure" they can get a web site up when for some unexpected reason, perhaps a simple oversight, they cannot.

**Very Important:** By default, IIS 7 running in Windows Vista does not allow web applications to create files in the Temp folder. When using IIS 7 in Windows Vista you must allow web applications to create files in the Temp folder. See the IIS 7 and Vista Notes section below.

Manifold IMS

All editions of Manifold System except for Personal Edition include the ability to publish a map for viewing on Internet through ordinary browsers. The Manifold Internet Map Server (IMS) is an integral part of Manifold System. Manifold IMS can publish maps, drawings, images or surfaces from .map files to Internet in conjunction with Microsoft’s Internet Information Server (IIS).

In addition to providing web sites that can be viewed with a browser, Manifold can also provide three specialized types of servers:
• **Manifold Image Server** - A Manifold Image Server is a generic type of image server for which drivers can be created and installed via the Manifold Image Server Interface (ISI) to enable usage within Manifold System clients. ISI drivers have been written for TerraServer, Google Earth, Yahoo Maps and Microsoft Virtual Earth, to name a few image servers. Manifold includes a built-in ISI driver to enable Manifold clients to acquire images on demand from a Manifold IMS installation, which can function for those Manifold clients as an image server. See the **Linked Images from Manifold IMS Web Sites** topic. Manifold is both a client and a server for the Manifold Image Server interface.

• **OpenGIS Consortium (OGC) WMS Server** - An OGC WMS server synthesizes images to deliver upon request to OGC WMS clients. Manifold is both a WMS server and a WMS client. See the **Linked Images from OGC WMS Servers** topic for information on using Manifold as a WMS client.

• **OpenGIS Consortium (OGC) WFS Server** - An OGC WFS server serves vector data to clients and can accept modifications to that data from clients. Manifold can function as an OGC WFS server for vector components such as drawings, themes and maps that contain drawing or theme layers. Manifold is a WFS server but is not a WFS client.

With Manifold IMS anyone with a Windows server and a DSL connection, cable modem with static IP or other full-time Internet connection with a static IP address can publish dynamic map projects to the web for the entire world to see. Organizational users can publish Manifold projects on their Intranets to provide convenient viewing of data through browsers.

There are four steps to using Manifold IMS to publish to the Internet:

- Use Manifold to create a project containing the map or other component to be published. The entire rest of this documentation describes the many capabilities of Manifold at your disposal for creating cool maps.
- Create the files required for a map server web page using File - Export - Web Page. This topic describes this step in detail.
- Use the created files within a web site. If the files are created directly within a folder in your IIS directory tree (such as C:\inetpub\wwwroot) the result of the File - Export - Web Page dialog will be an immediately "live" web site. This step is covered in the next topic, Creating a Web Site.
- Check to make sure the **IUSR_** account for the IIS machine (or, the **NETWORK SERVICE** account or **ASPNET** account if applicable) has access permissions to the web site you have created. Use Windows Explorer to view the security settings on all files involved in your web site. [This is an elementary part of setting up a web site within IIS and is really part of the third step above; however, it is so often overlooked by novice webmasters that Technical Support asks we include this as an explicit fourth step.]

Web programmers who understand ASP or ASP.NET technology can, of course, edit the files created by Manifold in the Export Web Page dialog to make their own customizations. Experienced web programmers can write their own ASP or ASP.NET files from the ground up, instantiating and working with Manifold objects as desired to create whatever Manifold-enabled web sites are desired.

Manifold exposes a very wide range of objects enabling the creation of rich web-based applications by programmers. There are many more capabilities made possible through customized programming, for example, display of terrains, than can be captured within the default templates within the automated Export Web Page dialog. Aspiring webmasters should study carefully the examples published by Manifold in the **Free Stuff** page on the manifold.net web page for ideas.

### Installation and Activation

Manifold IMS is an integral part of Manifold System. Manifold System must be installed on the machine that runs a Manifold IMS web site.

- During installation, the installation dialogs offer a choice to install Manifold System for anyone who uses the computer or just for the login in use. Manifold System must be installed for use by everyone who uses the computer in order for Manifold IMS to function.
- When Manifold is running as a map server it will not raise the Activation dialog. Before attempting to use the map server within Manifold you must launch Manifold interactively at least once to provide a serial number for preliminary installation, or a serial number and Activation key for permanent installation. See the Activation Keys and Serial Numbers topic.
- Note: after installing Manifold on a new IIS machine for the first time, do an **iisreset** command or reboot the system.
You cannot run Manifold interactively at the same time IMS is running, and you cannot run IMS at the same time that someone is using Manifold interactively on the same machine. If you are trying to use the same machine to run Manifold interactively and also to host an IIS / IMS web site, you must organize your activities so that an interactive session and IMS are not working together at the same time (or, one or both will not work). In such cases you must do an iisreset command after working with Manifold interactively before launching IMS, and you must stop the website and then do an iisreset command before launching Manifold interactively.

You can use a Manifold Runtime license to operate an IMS website. However, using a Runtime license requires more computer literacy because the Activation and administration of a Runtime license requires more advanced skills. Almost always, any serious application will deploy Manifold IMS using a Universal Runtime license because the price differential between a Professional Runtime license and a Universal Runtime license is slight and deploying with a Universal Runtime license assures that all extensions will be available in the future. Using a Universal Runtime license also eliminates the logotype that will otherwise be embedded within Manifold IMS displays.

Using the File - Export - Web Page Dialog

To publish a map page for use with Manifold IMS, open the component to be published and choose File - Export - Web Page to launch the Export Web Page dialog. This dialog creates a set of Active Server Page (.asp) pages or ASP.NET pages that can be used in a Microsoft IIS web site.

The Manifold project must have been saved at least once to enable the File - Export - Web Page command. A new project that has never been saved cannot be exported as a web page, since a physical .map project file must exist somewhere on disk for use by the web site.

The following instructions assume we have already created the Manifold project that will be published. Most users will have a working copy of the project in some development directory. They will then save a copy of the project to a directory on the server from which it will be published. The map server can then use the copy of the project for publishing.

To create a Manifold IMS web page:

1. Launch Manifold and open the .map project that contains the component to be published.
2. Open the component that is to be published.
3. Use File - Save to save the project .map file to the location on the server where you want the working copy of the .map file to be stored.
5. In the Export Web Page dialog choose a folder in which the web page files should be created. This could a temporary folder or it could be the default folder at C:\inetPub\wwwroot that will be used for the web page. The dialog will automatically expand a dot character “.” to the name of the current folder.
6. Specify other options as desired and choose OK.
7. Close the .map file and exit Manifold.
8. Use the created files within your web site as described in Creating a Web Site.

**Dialog Options**

- **Folder**: Name of the folder in which to create the new default.asp page. Use the […] button to the right of the Folder box to call a standard Windows browse dialog.
- **Template**: Choose a template to use to create the web page. Different templates will yield different site structures as well as different appearance styles.
  - **Standard**: Allows all options. Displays each new table in a new browser window.
  - **Standard (framed)**: Allows all options. Displays tables in a frame within the same browser window.
  - **Compatible with 4.x browsers**: Allows all options except the Info tool. Generates HTML compatible with older browsers. Displays tables in the same browser window, thus requiring a back command to see the map.
again.

Template selections also include ASP .NET versions of the templates. These should only be used when operating Manifold IMS in an ASP .NET environment. See the section at the end of this topic on using Manifold IMS in ASP .NET.

**Size**

Size of image to serve. The actual visualization of maps or layers within maps must be served to the browser as an image. Choosing this parameter chooses the size of image served. Any size may be specified, although usually settings are similar to aspect ratios seen in computer display monitors to provide pleasing proportions. Typical settings might be 320x240, 400x300, 640x480 to 800x600. Larger images will result in slower map server operation and slower downloads for viewers. Larger images will also require more of your Internet connection bandwidth.

**Embed description info**

If checked, write text properties into the .png image that is served in the web page using the component name as the Title property, the component's description text property as the Description property and the Software property set to the "Manifold System..." identification string displayed in the Help - About dialog. This option is a useful way of embedding copyright slogans: enter the copyright text or whatever other text you want embedded into the image into the component's Description property and it will be embedded into the .png image served. Virtually no one who saves your web site's image to their local machine will realize that you have, in effect, "watermarked" it with embedded .png text properties.

**Title**

Optional title string to use for the generated page. By default the title is set to the name of the component.

**Subtitle**

Optional subtitle string to use in the page. This string is often used to provide directions, such as "Click on a tool to choose it and then click in the map." When using the "Compatible" template, the directions might be "Click near the edges of the map to scroll in that direction."

**Copyright**

Optional copyright string to include in fine print at the bottom of the display. Set to a default copyright string by default.

**Queries**

If any queries exist in the project, they will be listed in this pane with a checkbox in alphabetical order. Check a query's checkbox to include it in a Queries pane on the web page. See the IMS Queries topic.

**Find tool**

Generates a control pane that allows searching for a particular value in a specified field. This control is for drawings and maps that contain drawings only. Note that the Find tool will list all fields in all drawings in a map that is published, excluding the ID field and intrinsic fields. To avoid confusing users, delete all undesired fields from drawings that are published. Objects found by Find will be shown in selection color.

**Hyperlinks**

Enable hyperlinks. If a drawing contains a field named URL (which may be any text type or URL type field) double clicking on an object will launch the contents of the URL field in a browser.

**Info tool**

Generates a toolbar button providing an information tool that displays fields of a clicked object or relevant information for a clicked label or pixel. The clicked object will be shown in selection color and a table will be displayed containing its fields. The Info tool will list all fields in all drawings in a map that is published, excluding the ID field and intrinsic fields.
<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Layers</strong></td>
<td>Generates a control menu that allows turning layers on and off. This control is available for maps only.</td>
</tr>
<tr>
<td><strong>Legend</strong></td>
<td>Include a legend in the web page.</td>
</tr>
<tr>
<td><strong>Location</strong></td>
<td>Generates a status note at the bottom of the display showing the latitude and longitude of the center of the current display.</td>
</tr>
<tr>
<td><strong>Logo</strong></td>
<td>Enabled in Enterprise Edition only. Uncheck to eliminate the &quot;Powered by manifold.net&quot; notice and hyperlink at the bottom of generated images.</td>
</tr>
<tr>
<td><strong>Scale</strong></td>
<td>Generates a status note at the bottom of the display specifying current scale in 1:xxx form. This control is for projected components only.</td>
</tr>
<tr>
<td><strong>Views</strong></td>
<td>Generates a pane containing a hyperlink for each named view. Named views (saved in the Views pane for that component) will appear under the names used for them in the Views pane. Available only if the component has views saved.</td>
</tr>
<tr>
<td><strong>Launch hyperlinks in new browser window</strong></td>
<td>If checked, when a user clicks a hyperlink a new browser window will be launched to display the link. Otherwise, the link will be displayed in the same window. The value of the option is saved into the <code>hyperlinksNew</code> entry within the <code>config.txt</code> file.</td>
</tr>
<tr>
<td><strong>Refresh linked components every ... minutes</strong></td>
<td>Automatically refresh every given number of minutes any linked component, such as a linked drawing or other linked component, which is used in the published component. Refreshing a linked drawing will update it from the controlling database table and refreshing a linked table will guarantee that it uses the latest values from the external table. Choosing a value of 0 will cause the project to be refreshed on every browser request. Do <strong>not</strong> choose 0 unless a very small number of users will be visiting the website.</td>
</tr>
<tr>
<td><strong>Include admin page</strong></td>
<td>If checked, create an administrative page for the web site in the form of a <code>default_admin.asp</code> page. The administration page shows the number of current sessions, the dates of the first hit, last hit and last reload, the total hits and a <strong>Reload</strong> button that can be used to force a reload of all <code>.map</code> files used by all sites hosted by this IMS server. If you create a <code>default_admin.asp</code> page for your IMS site, <strong>make sure</strong> to change access permissions on it so that ordinary Internet browsers cannot use it.</td>
</tr>
<tr>
<td><strong>Include Manifold Image Server interface page</strong></td>
<td>Creates an <code>imageServer.asp</code> page that allows Manifold clients to fetch images from this website using the Manifold Image Server interface via linked images. See the Linked Images from Manifold IMS Web Sites topic.</td>
</tr>
<tr>
<td><strong>Include OGC WFS interface page</strong></td>
<td>Create an <code>wfs.asp</code> file that allows OGC WFS clients to use this Manifold IMS web site as an OGC WFS server. Works with all web site templates. Turning on this option turns on OGC WFS server functionality via the resultant <code>wfs.asp</code> page.</td>
</tr>
<tr>
<td><strong>Include OGC WMS interface page</strong></td>
<td>Create a <code>wms.asp</code> file that allows OGC WMS clients to use this Manifold IMS web site as an OGC WMS server, supporting a variety of WMS protocol versions. Works with all web site templates. Turning on this option turns on OGC WMS server functionality via the resultant <code>wms.asp</code> page.</td>
</tr>
<tr>
<td><strong>Use custom server capabilities document</strong></td>
<td>Enabled only if the <strong>Include OGC WMS interface page</strong> option is checked. Allows specifying a custom XML capabilities document for the <code>wms.asp</code> file to use. If we enable this option we must provide a server capabilities document.</td>
</tr>
</tbody>
</table>
document written in XML using the schema published by OGC for such documents. Note that this document is not something that is created automatically by Manifold. If desired, we must provide a capabilities document. Such documents normally include information of interest to WMS consumers, such as, perhaps, the pricing and availability of a WMS service or other information.

Save proxy connection and file locations in the config file

By default, this option is turned off, to minimize the risk of leaking sensitive data such as passwords or physical paths used on the system. In some cases, such as when the exported web page uses components linked from remote servers, it might be necessary to turn the option on to supply these components with data required for correct function. See the topic on config.txt options for details on config.txt lines written.

Note: Attempting to export a web page that contains an image linked from a remote server or a map that includes such an image as a layer will display a warning message if this option has not been checked. This is a reminder to reconsider if proxy connection info and file locations will be required in the config.txt file.

Default Controls

The following buttons are created if their control is selected in the web page dialog. Examine the image file names used in the created web page to see the names of the files. These may be opened and edited to alter the appearance of the buttons, if desired, or other graphics files may be substituted by editing the created web page.

- Zoom in - Zoom to a closer view.
- Zoom out - Zoom to a view farther away.
- Zoom Box - Zoom into the box drawn with the tool. Created only with Standard and Standard framed templates.
- Zoom to Fit - Zoom to fit entire data set in current view.
- Center - Center view at clicked location.
- Info Tool - Show data fields for object. Shows all fields in all drawings in a map that is published, excluding the ID field and intrinsic fields
- Expand Heading - Open a tool such as Find.
- Contract Heading - Close a tool such as Find.

Controls used in sites generated with the Compatible template are simpler. The map is panned by clicking at different locations in the map with the mouse cursor. Other controls are simple links. Webmasters should experiment with the map server by creating the same web site using different templates to see the different pages created.

Files Created

The Export Web Page dialog creates several main files and one directory containing additional files that are used in the web site to create the web page seen by browsers. Files are created by default in the folder specified by the Tools - Options - File Locations - Web Pages option, by default, C:\Inetpub\Wwwroot.

Files created by the Export Web Page dialog include:
default.asp  Main .asp page. Calls other pages with or without frames depending on the template used.

default.css  Cascading style sheet file specifying fonts and other styles.

default_admin.asp  Administration page that shows the number of current sessions, the dates of the first hit, last hit and last reload, the total hits and a Reload button that can be used to force a reload of all .map files used by all sites hosted by this IMS server. This administration page is created if the Admin page box is checked in the Export Web Page dialog.

If you create a default_admin.asp page for your IMS site, make sure to change access permissions on it so that ordinary Internet browsers cannot use it.

config.txt  Configuration information. Provides path to the .map file to be used, the name of the component within the .map file to be published, x size in pixels of image, y size in pixels of image, an optional title and an optional subtitle. The path to the .map file to use is taken based on whatever .map file was being used when the Export Web Page dialog was used. The config.txt allows setting some other options, such as click radius, manually.

images  A directory containing image files in .png format that are used to create toolbar buttons and controls.

imageServer.asp  An interface page that allows Manifold clients to fetch images from this website using the Manifold Image Server interface via linked images. See the Linked Images from Manifold IMS Web Sites topic.

map.asp, mapview.asp, tableview.asp  Accessory .asp files that may be created by some templates that are used to create tables and manage the map view displayed.

wfs.asp  An OGC WFS interface file to be used by WFS clients, if the OGC WFS interface page option has been enabled.

wms.asp  An OGC WMS interface file to be used by WMS clients, if the OGC WMS interface page option has been enabled.

The .asp files work in concert to generate the HTML stream that is sent to a browser by IIS to display the map image, toolbar, status bar, menu panes and tables that result from queries. Scripting code within the .asp files calls the map server to generate the correct view. Toolbar buttons and other controls are drawn using images stored in an images subdirectory.

Default Configuration

Optional templates used within the Export Web Page dialog will create different pages. The configuration seen in the browser window depends on choices made in the dialog. The illustrations shown below use the Standard template, which is designed for use with newer browsers. The Compatible with 4.x browsers template uses a simpler style. TheCompatible template provides all functions of the Standard templates except the Info button.
The default layout positions a toolbar above the main view. Buttons allowing zooming and positioning are in the toolbar.

Optional tools such as the Find tool or Legend are positioned in menu bars to the right.

Clicking on the down arrows in the menu headings expands the tools.

Permissions

Manifold documentation assumes the user is working with a modern Windows release such as Vista, Windows Server 2003 or XP using the NTFS file system.

Manifold must be installed for everyone to use on the machine.

The .map file used with the web site must have security permissions such that the IUSR_account used for Internet browsers has read access (or, the NETWORK SERVICE account or ASPNET account if applicable). This account begins with "IUSR_" and ends with the machine name. For example, if our machine is called MAPSERVER the account will be called IUSR_MAPSERVER.

Very important: In 32-bit and 64-bit versions Windows Server 2003 and in 64-bit Windows XP the account used for Internet visitors is not the IUSR_account as with earlier Windows versions. Instead, the NETWORK SERVICE account is used. When ASP.NET is employed the ASPNET account is used.

Windows Server 2003 and 64-bit Windows XP operators should understand the NETWORK SERVICE account is meant in all cases where this documentation discusses the IUSR_account.

Web pages operating in an ASP.NET environment may require different permission settings than those operating in an ASP environment. For example, the default login used by ASP.NET is ASPNET, while the default login used
by ASP is either the IUSR_ account or the NETWORK SERVICE account. When running in ASP.NET, the .map file and all other data being accessed from the ASP.NET environment must be readable by the ASPNET user account.

ASP.NET operators should understand the ASPNET account is meant in all cases where this documentation discusses the IUSR_ account.

**Very important:** Failure to pay attention to permissions and to grant the IUSR_ account (or, the NETWORK SERVICE account or ASPNET account if applicable) read permissions to the .map file being used is the number one technical support call for IMS. If you experience difficulty browsing an IMS web site, save yourself a SERVICE Very important:discusses the account names that have access to that file. If it does not, make sure you have important:discusses the permissions.

**Important:** Make sure to use Windows Explorer to check permissions (and not the IIS management applet) and make sure you have Administrator privileges so you can view and change security settings for other accounts.

**Important:** If you use Manifold in interactive mode on the server machine to alter the .map file or even to open it, make sure to use Windows Explorer to check permissions on the IUSR_ account (or, the NETWORK SERVICE account or ASPNET account if applicable). Opening the .map in Manifold may reset its permissions to something other than having the read permissions for the Internet access account. Saving a .map file with Manifold will preserve file permissions on Windows 2000 and later systems, but this is always something worth checking.

If any linked drawings or table relations referring to external databases are used in the project that is published, the IUSR_ account must have read access to the tables that control those linked drawings or relations. In some cases, such as when .mdb files are used, the IUSR_ account must have write permissions to the folder in which those files are located, since the map server process must create a lock file as is required when working with .mdb.

Take time to think through all permissions issues. The Internet access account must have permissions to get to all the data that is used in the web site. For example, suppose our IMS project uses a linked image. Because linked images use data from the Data Cache folder (the default), we must also be sure to give the IUSR_ account (or, the NETWORK SERVICE account or ASPNET account if applicable) read permissions to access all files in that folder.

Windows administration errors that result in the IUSR_ login (or, the NETWORK SERVICE account or ASPNET account if applicable) not having read access to the .map file are one of the most frequent reasons why an IMS web site will not launch. If the IMS web site does not launch, use Windows Explorer to check the permissions on the .map file and on the folder in which it is located as well as on all other files and folders if they are involved, such as by use of a linked drawing or table relation. See the discussion in the Creating a Web Site topic for more information.

Please read the above Permissions paragraphs before contacting Tech Support with questions about IMS pages not working. Don't waste developer support incidents on easy stuff like this!

Geocoding SQL extensions will not work with Manifold IMS unless the US streets geocoding database is installed within the Manifold application installation folder (normally, C:\Program Files\Manifold System). Therefore, the US streets geocoding database should be installed in the Manifold application installation folder on machines on which Manifold IMS operates. The default installation path for the Manifold Geocoding Data product is C:\Program Files\Manifold System\GCDB, which satisfies the above criteria if Manifold has also been installed to the default path.

**Note:** Security permissions are part of modern Windows editions such as Windows 2003, Windows 2000 and Windows XP. However, security options are available within Windows 2000 only if the disk drive is formatted using NTFS. If a disk drive has been formatted using FAT32 (as is possible, for example, if someone has arranged a dual boot system that can be booted either as Windows 98 or Windows 2000) then security permissions will not be available.

Since FAT32 does not provide any security options all files and folders located on FAT32 volumes can be read by and written to by anyone (including IUSR_). This, of course, makes it entirely possible to host IIS and IMS on a FAT32 drive, albeit without any security options.

**IIS 7 and Vista Notes**
By default, IIS 7 running in Windows Vista does not allow web applications to create files in the Temp folder. When using IIS 7 in Windows Vista you must allow web applications to create files in the Temp folder. The easiest way to do this is to run the following commands from the Command Prompt with Administrator privileges as follows:

To allow web applications in IIS 7 and Vista to create files in the Temp folder:

1. Click on the Windows Start button and then use the Start Search box.
2. Enter "command" to begin searching for the Command Prompt program. A list of matches will appear.
3. Right click onto the Command Prompt program and choose Run as Administrator. Enter the Administrator password.
4. This launches the Command Prompt window with Administrator privileges. Enter the following two command lines, pressing Enter after each line (enter each line as a single line):

   icacls %windir%\serviceprofiles\networkservice\AppData\Local\Temp /grant Users:(CI)(S,WD,AD,X)
   icacls %windir%\serviceprofiles\networkservice\AppData\Local\Temp /grant "CREATOR OWNER":(OI)(CI)(IO)(F)

5. Restart IIS using IISRESET and then close the Command Prompt window.

MapPoint and IMS

MapPoint geocoding will normally not work from IMS unless we map anonymous Internet connections to a user account that has more enhanced permissions than the default Internet access IUSR_xxx account (or, the NETWORK SERVICE account in Windows Server 2003).

To enhance permissions for the Internet access account:

1. Create a regular user account with default permissions that is a member of the Users group.
2. Open Control Panel - Administrative Tools - Internet Information Services (or the equivalent dialog in your version of Windows).
3. Right click the folder that contains the published website, select Properties, switch to Directory Security and click Edit under Anonymous access and authentication control.
4. Set the user account used for anonymous access to the account created in step 1 above.

See the Geocoding with MapPoint topic for use of MapPoint as an auxiliary geocoder.

How the Map Server Operates

The files created by File - Export - Web Page are placed in a directory within the IIS web site hierarchy, and an IIS application is created using that directory (see Creating a Web Site for details). On the first call to the map server, the map server reads the config.txt file to see what .map file is to be used, the component within the .map file to be published, the size of the map to generate and text to use for tile, subtitle and copyright lines.

When a user browses into the default.asp page, IIS executes the code in the default.asp page (which invokes the other .asp files) to create a web page that is served to the user's browser. Part of the code in the .asp files involves relatively simple HTML that is easily modified. For example, the table structures used to provide the title string or the copyright string are implemented using elementary HTML that may be modified to customize the appearance of the page as desired. Other parts of the code are more sophisticated, client-side or server-side scripting commands that manage the programmatic task of detecting and reacting to user commands and updating the display.

The actual image displaying the map is created by Manifold IMS in a call to the map server. Manifold IMS opens the .map file and works with the component specified to generate a view of the component fitting the zoom and pan commands issued by the user. The map server reads the config.txt file in the same folder as the default.asp file to know how to configure the display. The final view is rendered as a .png image (or other image format as settable in the IMS config.txt file options) and served by the map server to the browser within the stream of HTML commands created by IIS's processing of the .asp pages.
From the above description it should be clear that when we use File - Export - Web Page to create a default.asp file and attendant files we are not really exporting the component into a different form. Instead, we are creating a set of instructions that are executed by IIS to call Manifold through the map server to use the published component to create web views of our data. Note that when the map server is operating the .map file is in use and cannot be opened by Manifold.

To allow simultaneous editing of the .map file, make a copy of the map file and save the copy as the .map file that is published. See Creating a Web Site for additional operational tips.

When IMS operates updating the .map file (say, by copying a different version into the same location) will not update what is served to the web. To change the .map file dynamically the .map file must be reloaded into the MapServer object either using the Reload method programmatically or by using the administrative page to force a reload manually by pressing the Reload button. Reloading will cause a reload of all .map files used on the server hosting IMS.

For expert usage, IMS opens .map files for both reading and writing. This allows doing sophisticated analysis that involves creating temporary components, such as running queries the text of which is supplied by the user.

**Operation as a WFS Server**

The Manifold OGC WFS interface can be used with vector components, such as drawings, themes and maps that contain drawing or theme layers. The Manifold OGC WFS server supports HTTP POST as well as altering data within served component or other components.

**Operation as a WMS Server**

When an OGC WMS client connects to the wms.asp page created when the Include OGC WMS interface page option is selected, Manifold serves the image tiles requested by the client. Just as Manifold in regular IMS operation provides an image that is a view of whatever map layers are being displayed, the WMS server provides an image to the specifications requested by the client of whatever layers are the map and requested by the client. The Manifold OGC WMS server respects any transparent pixels in images used, rendering the resultant images with transparent pixels as well.

Suppose, for example, we create a map that shows member countries of the European Union. We export it to a folder called C:InetPub\wwwroot\euro and we also check the Include OGC WMS interface page option. Suppose our Windows server is set up to serve the domain name mydomain.com. A visitor browsing to http://www.mydomain.com/euro or http://www.mydomain.com/euro/default.asp with their web browser would get the IMS version of the site for their browser. A visitor connecting with an OGC WMS client (such as Manifold) would use a URL of http://www.mydomain.com/euro/wms.asp to connect to Manifold running as a WMS server.

See the Linked Images from OGC WMS Servers topic for a discussion of WMS clients operation.

**Enterprise Edition and OGC WMS Server Applications**

Enterprise Edition is strongly recommended for OGC WMS server applications because Enterprise Edition allows suppression of the "Powered by manifold.net" logo that is added to all IMS images by Professional and Professional Runtime editions. Each tile transmitted to a WMS client will include the logo at the bottom of the tile. When using Enterprise Edition we can turn off the logo. Note that deploying a Manifold IMS application using a Universal Runtime license in effect installs Enterprise Edition on the server.

**Projections and OGC WMS**

There is no standard way for an OGC WMS server to describe the coordinate systems (projections) it supports that covers all coordinate systems supported by Manifold. There are several semi-standard ways using schemas to describe the coordinate system used by a WMS server that cover various subsets of coordinate systems.

When used as an OGC WMS client, Manifold supports all schemas for describing coordinate systems that have been uncovered by manifold.net, no matter how many or how few OGC WMS servers that currently exist actually use them. Given the tendency of OGC to come up with new "standards" that are never used to any significant degree, it is always possible yet another schema may be introduced in the future that is not recognized by Manifold.
as an OGC WMS client; however, we feel confident that the current edition of Manifold when used as an OGC WMS client covers all such schemas either proposed or in actual use.

In contrast, when used as an OGC WMS server, Manifold only supports EPSG codes. EPSG is arguably both the richest and the most widely used schema within the OGC community. Manifold running as a WMS server only supports EPSG mostly to avoid confusing various non-Manifold WMS clients, some of which, unlike Manifold, cannot auto-adapt to a variety of coordinate system encoding schemas.

As rich as it is, the EPSG schema only supports a fraction of the coordinate systems that can be used in Manifold. If the coordinate system used by the component served by Manifold can not be expressed using the EPSG schema, that is, if it does not have an equivalent EPSG code, then the Manifold OGC WMS server reports the coordinate system as latitude / longitude and will re-project the data for that component on the fly into Latitude / Longitude. For large components that will have a terrible impact on performance.

Obviously, it is essential that any webmaster setting up a Manifold application as an OGC WMS server should check the projections used by the components being served and should make sure that they are coordinate systems that can be expressed using the EPSG schema.

The current list of coordinate systems which have EPSG codes may be obtained free of charge from the following link, clicking the link to the epsg-v69.zip or subsequently published file:

http://www.epsg.org/CurrentDB.html

If all we are doing is using Manifold as an OGC WMS client we don’t need to worry about the above because any schema used by the OGC WMS server we access as an image server will be recognized by Manifold.

Operation as an Image Server

Manifold System can connect to image servers of various kinds (such as TerraServer or Google) to automatically fetch images that cover a desired region to create a linked image. Manifold can also function as a Manifold Image Server so that Manifold users can fetch linked images from that server using the Manifold Image Server interface built into Manifold.

When a Manifold System client connects to the imageServer.asp page created when the Include Manifold Image Server interface page option is selected, Manifold serves the image tiles requested by the client. Just as Manifold in regular IMS operation provides an image that is a view of whatever map layers are being displayed, the Image Server page provides an image to the specifications requested by the client for the region of interest.

See the Linked Images from Manifold IMS Web Sites topic.

The Administration Page

If desired, the Export Web Page dialog can create an administrative page if the Admin page box is checked.

The administrative page is a default_admin.asp page that is created in the same folder as the rest of the IMS web page. The administration page shows the number of current sessions, the dates of the first hit, last hit and last reload, the total hits and also provides a Reload button that can be used to force a reload of all .map files used by all sites hosted by this IMS server.

The administrative page makes it easy to update a .map file with a newer version without having to restart IIS. To do this, edit a copy of the .map file and save it under a similar name, making sure that after it has been saved permissions are as required. For example, if we are using myfile.map with the map server we could edit a copy and save it under the name myfile2.map. Next, change the config.txt file to use myfile2.map instead of myfile.map and then navigate to the administrative page and press the Reload button.

Because the administrative page's .asp file will be created in the same folder as the default.asp page that powers the site, it will also by default be created with the same access permissions as the default.asp page. That is, anyone going to http://(sitename)/default_admin.asp will be able to operate the administrative page. They will be able to see your web page's statistics and, more importantly, they will be able to force IMS to reload all .map files used by all IMS pages on that server by simply pressing the Reload button. Since a reload takes time, it is not a good idea to allow any visitor to force a reload however often they want to push the Reload button.
**Very Important:** If you create a `default_admin.asp` page for your IMS site, make sure to change access permissions on it so that ordinary Internet browsers cannot use it. To do this, in Windows Explorer right click on the `default_admin.asp` file, choose Properties and choose the Security tab. Remove all entries for allowed user names except Administrator or whatever other login name you wish to allow access to the administration page. Apply the settings and press OK.

Neglecting this important step will not create a security breach in that hackers will be able to steal your data, but it will allow a malevolent user to effectively shut down your map server by repeatedly causing it to `Reload.map` files.

### Layers

When a map component is published via the map server, the Layers control menu option is enabled. Choosing the Layers option in the Export Web Page dialog will add a Layers menu to the web page.

![Layers](image)

Users can check or uncheck the layers they wish to appear in the map window and then press Apply.

Layers will appear using the same names they have in the map. Therefore, one should choose wisely the names of such layers. Rename any drawings or other components so their names will be useful to people browsing the web site. For example, if creating a map of Mexico one may import a drawing of Mexican railroads from a file called `me_railroads.mfd`. This will be imported into Manifold using the name `Me_railroads Drawing`. It would wise to rename such a drawing to "Railroads" so that when it appears as a layer in the map and thus in the Layers control in the web page it is also named "Railroads".

When a map is published using Export Web Page the map server shows all layers that are in the map in the Layers pane.

### Layer Restrictions

Layer restrictions in maps are respected in that both hyperlinks and the Info tool in default web page templates will work only on layers that are clickable. To disable hyperlinks or the Info tool in a layer, open the map, right click onto that layer's tab and choose Restrictions and uncheck the clickable box. Interactive selection and editing layer restrictions are not factors in IMS web pages so these layer restrictions are ignored. See the layers pane topic for more information on layer restrictions.

Use of the clickable layer restriction is very important when creating a web page in which hyperlinks or the Info tool will be used. With such web pages there is usually a layer that contains the objects of interest while other layers serve as background layers only. The classic example is a web page that shows dealers that are near some location. The user would like to double click on a dealer point icon to launch a web page with information about that dealer, and the designer of the product wants to make sure that an accidental click on a background map layer, such as a state or nearby road, is ignored.

Accomplish this by having the layer containing the dealer points be clickable with the other layers not being clickable.

### Click Accuracy

The default click accuracy in Manifold web pages is 4 pixels. Clicking within four pixels of the center of a point will be the same as clicking directly onto the center of the point. This is normally a good balance between not requiring too much manual dexterity of the user while allowing points to be reasonably close together but separately clickable in dense displays.
To decrease click tolerance, add a line such as the one below to the config.txt file:

```
clickAccuracy = 8
```

The above line increases the clicking tolerance to 8 pixels, so that a mouse click within eight pixels of the center of a point will be the same as clicking directly onto that point.

**Control Layers with Zoom Ranges**

Many Manifold IMS pages will not use the Layers control but will publish all layers in a map as a single image into which users can zoom. In such cases we will often want layers to appear and disappear automatically as users zoom in to or out of the map. For example, as users zoom into a map we might want major roads to appear and as they zoom further into the map we would like detailed streets to appear. Control the appearance of layers using Zoom Ranges.

**Automatic Panning and Zooming**

When objects are selected with the Find tool or by a query (if the query includes the ID field), Manifold IMS will automatically pan the display to the centroid of the selected set of objects. Zoom will be adjusted to zoom to fit to the selection.

If only one object is selected and it is a point, or if several points are selected that are coincident, Manifold IMS will center the point on screen and will adjust zoom so that the display shows approximately 1/100th of the total data set. The result is usually a good level of zoom for displaying a single point within an entire data set.

**Customization**

See Creating a Web Site for notes on customizing the configuration created by File - Export - Web Page. See the Manifold web site for examples of customized IMS applications programmed in ASP or ASP .NET.

**Manifold IMS and ASP .NET**

This is an advanced subject for those webmasters who will be operating Manifold IMS in a Microsoft ASP .NET environment. To run Manifold IMS in ASP .NET we must choose an ASP .NET version of the templates in the Export Web Page dialog. You must have ASP .NET enabled in your IIS installation (easily done using IIS Manager) if you want to use ASP .NET.

The Export Web Page dialog includes ASP.NET versions of the Standard, Standard framed and Compatible templates. These must be used when operating in an ASP .NET environment.

**Automatic Refresh**

If a published component is a linked component (such as a linked drawing or other linked component) or if a map is published that includes any linked components as layers, we can turn on the Refresh linked components every ... minutes option and specify a reasonable number of minutes between refreshes. This will assure that the linked components displayed on the web site will be updated on a reasonably recent basis from the data sources from which they are linked.

Refreshes using this option apply only to that component published on the web site. For example, if the web site shows a linked drawing, this option will only refresh that drawing. If the web site shows a map incorporating as layers three linked drawings and one linked image but the project file in use also includes seven more linked drawings which are not used in the map, this option will refresh only the three drawings and the one linked image that participate in the map.

Automatic refreshes apply only to components in use in order to maximize performance, as there is normally no point to wasting machine time to refresh components that are not displayed. If for some reason we would like to refresh a particular linked component within the project other than those used in the web site, we can do this programmatically using Component.Refresh. If we want to refresh all linked components in the project file, we can easily do this using Document.RefreshAllLinked. These tasks are accomplished programmatically since it is presumed that anyone with the technical expertise to desire such customization also will have the technical wherewithal to be able to edit or write ASP or ASP .NET code.
The performance cost of refreshes varies widely depending upon the nature of the linked components and the nature of the data source providing data to the linked component. Manifold has immense flexibility and many options for creating linked components, which can get their data from many different sources.

A linked drawing, to name just a few examples from many possible, could be linked from a local table in the project, from a query or from an external data source. The external data source could be something as simple as a .csv text file or it could be something as sophisticated and powerful as an Oracle Spatial server. The communications path between the linked component could also vary. It might be as simple as a component within the same project file in the same server, or it could be a relatively slow connection via Internet to some server sitting half-way around the planet.

The data involved might be a small amount of data or a very large amount of data. Some technologies that might be used for linked components do better at providing changes (if any) when large amounts of data are involved while other technologies can be very inefficient, requiring a complete transmission of all data just to be sure any changes are incorporated.

None of these factors will surprise the experienced web developer, who is used to the notion that linking content into web pages from a variety of disparate sources and technologies exposes one to the vagaries of performance attendant to such varied sources. But the newbie web developer should think carefully about the architecture in play and not promiscuously jump to refresh frequently in a "just in case" manner to cover up any lack of understanding of how the project is put together.

It is also important not to be lured into false promises of accuracy or timeliness by refreshing a web site more frequently than makes sense given the nature of the data and the architecture of the application. For example, consider a web site that shows the positions of ocean buoys reporting weather data. The key component is a drawing linked from a geocoded table resident on some NOAA server. The geocoded table on the NOAA server is updated using some unknown software that acquires the locations of the buoys as reported by GPS devices and puts them into the table.

It doesn't make sense to refresh such a linked drawing every minute. Buoys drift very slowly to begin with, the GPS devices are not particularly accurate and ultimately it really doesn't matter for large scale weather observations over thousands of miles of ocean to know the location of a particular buoy give or take a kilometer or so. For that matter, we probably don't know how often the NOAA table is updated. Given these factors it could well be that some webmasters would choose to refresh their linked drawing every 60 minutes or so, while some more "gung ho" operators might choose a shorter interval, such as every 10 or 20 minutes.

It's true that if we don't have very many buoys (quite likely, only a few dozen and not tens of thousands) the cost of a refresh will likely be so low that it will be unnoticed even if done every minute. However, we should keep in mind that the data connection to the NOAA server might be slow even if only a small amount of data is exchanged. Therefore, it is better to think through the situation and choose a sensible refresh interval than to simply give up on thought and choose one minute "just in case."

Suppose our application works with more rapidly moving objects, such as a web site that reports the locations of vehicles, where waiting ten or twenty minutes between refreshes would give significantly inaccurate results? There are two strategies to follow in this case.

First, some vehicle tracking applications are intrinsically low resolution. Not all vehicle tracking applications provide reasonably real-time data into a database table that can also be accessed in real time. Other applications, such as the location of trucks in long-haul trucking services spanning an entire state or an entire continent, really don't care about the location of a truck give or take one or two kilometers. In such cases, we simply set the refresh interval to one minute and let it go at that. This is a good solution in such cases because such applications tend to have many more units being tracked. We usually don't need to track thousands of trucks in a small region, where it might matter how far the truck moves in one minute. Instead, we usually have to track thousands of vehicles only on a very large scale, such as over an entire state, where the precise position give or take a mile doesn't matter.

Second, for those higher resolution applications requiring faster response we can always use Component.Refresh programatically to refresh as fast as we would like. However, here too it pays to use common sense: if there is a latency lag in the entire communications chain of plus or minus ten or twenty seconds between the movement of a vehicle and the ultimate availability of that changed position to the server running Manifold, then it doesn't make sense to refresh every second. In any case we will be looking at the approximate position of a vehicle and not some exact position accurate to one meter.

Manifold deliberately uses minutes as the unit of measure in the automated refresh setting to force web developers to explicitly think about their web site's implementation and to avoid the tendency of newbies to
inappropriately pick the shortest refresh interval possible. A time scale measured in minutes is fine for lower resolution or large scale applications. Applications that need a refresh in seconds require greater professional attention which tends to come automatically with the enhanced technical skill required to use Component.Refresh programmatically.

Other HTTP Servers

Although Manifold IMS is aimed at use in a Microsoft IIS environment, one can write web pages to use IMS under any HTTP server provided the execution environment can instantiate and work with COM objects. The default templates for IMS create .asp files that are intended to be used within a Microsoft IIS environment; however, nothing prevents a user from adapting these files or writing their own script files or web pages for use with an HTTP server other than Microsoft’s IIS. This assumes, of course, that the non-IIS HTTP server being used is running on a Windows system since Manifold System runs only under Windows operating systems.

Map Server FAQ

What is the licensing for the map server? Each machine running IIS and serving from the map server must have a Manifold System license installed. Only one Manifold license is required per machine even if multiple sites are served from that machine. Any Manifold license except Personal Edition (which does not include Manifold IMS) is OK. Most users will develop Manifold IMS applications using a full license and then deploy using a runtime license, usually a Universal Runtime license.

How about generated images? That depends upon the licensing structure of the data you are using. Manifold's license permits you to serve unlimited images of any kind created from the Manifold data sets you use for no additional charge. With public domain data there are no licensing issues either. Some vendors may or may not require more restrictive licenses as a condition of selling you proprietary data.

I don't have a full time Internet connection… how do I use the map server to publish images? If you do not have a full time Internet connection you must install such a connection or find someone who has such a connection. Or, you could hire time from an ISP using a managed server that provides you with a virtual machine, like having your own machine connected full time to the web, but managed for you by the ISP.

How do I publish my maps to a web site I have at my ISP? Your ISP must install Manifold System on the machine that hosts your web site and both your .map file and generated web pages must be uploaded to that machine. Most people will do so via a managed server or co-location plan with the ISP. Because the map server runs extremely well on modern PCs that anyone can afford, another possibility is to install a full-time Internet connection (like a cable modem or DSL line) and run the server on your own machine. Even a relatively low-speed direct connection like a 384K uplink can handle thousands of users per day. Another option is to use an Application Service Provider (ASP) or consultant who can set up and operate your Manifold web page for you.

How do I customize the appearance of the generated web page? Change the HTML in the default.asp file and the style settings in the default.css style sheet. Advanced users can tinker with the scripts as well. Changing table structure, background colors and other design elements within these files can create radically different appearances. Add your HTML to customize the page with user instructions, titles and other texts.

Can I program the map server? Yes. By writing customized ASP or ASP.NET files one can create very sophisticated interactions with the project through the map server. This is an advanced topic that is described via the programming documentation in Help as well as with examples in the Free Stuff page on the manifold.net website.

How do I remove the small "Powered by manifold.net" text and hyperlink to the manifold.net IMS page? In Manifold System Professional Edition the logo cannot be removed since it is part of the view generated by the map server itself and is required to be present by the End User License Agreement for Professional Edition. In Manifold System Enterprise Edition the "Powered by manifold.net" text and hyperlink may be removed by unchecking the Logo box in the Export Web Page dialog. This is also possible, of course, when using a Universal Runtime license to deploy a Manifold IMS application to a web server.

Can I use Manifold IMS to publish more than one map from the same server? Yes. The Manifold map server can use more than one project at a time using different .map files and can even publish more than one component at a time from the same .map file. See Publishing Multiple Pages.

How do I run the map server in my cluster / web farm? The map server bundled with Manifold System does not automatically cluster across systems. A simple approach to distributing load between multiple servers is to use a round-robin proxy to distribute clients among your servers.
What are the browser requirements? Any modern browser can view pages generated by the map server. The browser must be able to execute scripts and to display .png images (or other image format specified in the IMS config.txt options). All "4.x" and later IE browsers may be used.

Can the map server run with HTML servers other than IIS? Yes. One can write web pages that use IMS under any HTTP server provided the execution environment can instantiate and work with COM objects. However, such usage is not supported by manifold.net since the main development environment and all default templates are targeted at IIS.

What is the IUSR_xxx account? Because it does not make sense to allow everyone on Internet to have free access to do whatever they want with our web server, Microsoft has thoughtfully provided an IUSR_xxx (xxx is the machine name) login that is automatically assigned to visitors interacting with the web server via Internet. Microsoft also has had the foresight to restrict the IUSR_xxx account so it cannot rampage around in our machine.

Very important: In 32-bit and 64-bit versions Windows Server 2003 and in 64-bit Windows XP the account used for Internet visitors is not the IUSR_xxx account as with earlier Windows versions. Instead, the NETWORK SERVICE account is used. When ASP .NET is employed the ASPNET account is used.

Windows Server 2003 and 64-bit Windows XP operators should understand the NETWORK SERVICE account is meant in all cases where this documentation discusses the IUSR_xxx account.

Web pages operating in an ASP.NET environment may require different permission settings than those operating in an ASP environment. For example, the default login used by ASP.NET is ASPNET, while the default login used by ASP is either the IUSR_xxx account or the NETWORK SERVICE account. When running in ASP .NET, the .map file and all other data being accessed from the ASP.NET environment must be readable by the ASPNET user account.

ASP .NET operators should understand the ASPNET account is meant in all cases where this documentation discusses the IUSR_xxx account.

When we set up an Internet application by putting files in the InetPub/wwroot folder hierarchy, by default visitors get reasonable IUSR_xxx account access to those files because Microsoft has set up those folders with inheritable permissions that grant the IUSR_xxx account access within those folders. However, if we want to use files that are not located in that folder hierarchy, we must explicitly grant permission to the IUSR_xxx account to read or execute such files. For example, if our website contains .asp files that instantiate an object that uses a .map file located somewhere else on our machine, we must make sure that the IUSR_xxx account can read the .map file.

Can Manifold IMS run on a Linux system? No. Manifold System runs only on Microsoft Windows operating systems.

How do I meet other developers working with IMS? Post on the Georeference forum. Visit the Forum page on the manifold.net web site or simply go directly to forum.manifold.net - if you are doing serious work with IMS, make sure to visit and participate in the forum where IMS programmers from all over the world meet to discuss IMS and exchange tips and advice.

Promote Your GIS Work

In just about any enterprise one must justify the resources invested into organizations and activities. Showing the tangible results of GIS work by publishing a cool IMS web site can be a spectacularly effective way of showing off your GIS work within your organization. It's easy to show off your GIS work on a web site that anyone with a browser can access from the convenience of his or her desktop. It is much easier to demo your work on a web site than it is to drag people one by one in front of a machine to show them a GIS demonstration.

Even if your mission does not require publishing a map-enabled website to Internet, it is usually a great idea to put up one or more IMS web sites on your organization's internal network just to make sure you get plenty of support within your organization for your GIS work.

If you are running a GIS business that could benefit from additional clients, running one or more IMS web sites that demonstrate your work is a great way to advertise your capabilities. Creating an IMS web site that provides some practical benefit for the industry you serve is a great way to generate links to your site. One good site that draws links from industry ezines and other sites can be worth tens of thousands of dollars in advertising per year.

Host Other Sites
Companies are now launching business plans to be Application Service Providers ("ASPs"), that is, to get into the business of providing hosting services over the Internet including the provision of GIS maps published over the web. At the present writing prices for such services are very high with startup fees of $1500 and recurring charges of $1500 per year for Internet map services that serve only 100 sessions ("hits") per day. Almost any computer that runs Manifold can host dozens of map server web sites at such low hit rates.

If you have a reasonably fast Internet link, IIS and Manifold System, you could provide hosting services for a fee to other Manifold users who do not want to host their own Internet server. Hosting Internet map server sites for government groups and other organizations is a great way to build a GIS consulting business as well, since such organizations often will want assistance in the preparation of the .map file used with the map server.

### About .PNG Format

The images served to the browser within the map window are created by Manifold IMS by default in .png format (other image formats may be specified in the IMS config.txt options). "PNG" stands for "Portable Network Graphics" and is also known recursively as "PNG is Not GIF".

.png (pronounced "ping" by .png experts) is a format originally developed to replace .gif. The .png format is lossless like .gif but is more efficient than .gif. In addition, .png can save alpha channel transparency. Although .jpg and .gif are more frequently used on the web as a matter of inertia, in simple form (without transparency) .png is universally supported by all modern browsers. .png images therefore can be used anywhere in a web page just like a .jpg or .gif.

Manifold uses .png because it is more efficient than .gif and suffers no loss of information in compression as does .jpg. By using .png Manifold is able to save at least 10% bandwidth as compared to using .gif. Since bandwidth in most sites is precious the simple use of a more efficient format allows at least 10% more simultaneous users at a given hit rate that could otherwise be accomplished with the same hardware and ISP service levels.

.png is also better than .gif because it is not subject to moronic patent threats from Unisys. See Just Say No to GIFs for a quick history on the threat to .gif usage from Unisys.

### Security and Access Permissions

See the Creating a Web Site topic for a discussion of security issues and access permissions.

### Troubleshooting

Manifold cannot run interactively with the usual user interface at the same time Manifold is functioning as a map server. All interactive sessions must be shut down before IIS can call upon the map server to serve map pages to the web. Conversely, if the map server has been run Manifold can not run interactively until the map server has been unloaded either via an iisreset command (the usual method) or by rebooting the system.

See the troubleshooting topic Problems with the Internet Map Server for detailed checklists of what might be wrong. Check to make sure that you are not running Manifold System Personal Edition, which does not include the map server. You must be running Professional Edition or greater to have the map server.

The number one problem with IMS reported to tech support is that users neglect to add the IUSR_ account (or, the NETWORK SERVICE account or ASPNET account if applicable) with access permissions to the .map file in use. Using Windows Explorer (do not just depend on the IIS management console or other server management console), right click on the .map file, choose Properties and verify in the Security that the IUSR_ account (or, the NETWORK SERVICE account or ASPNET account if applicable) for the system has necessary read and execute permissions.

### Note

You must be running Manifold System Professional Edition or greater to have the capabilities discussed in this topic. All Manifold editions except Personal Edition include the map server. If you are running Manifold System Personal Edition you will not have the capabilities discussed in this topic.
You must also have your Manifold license activated. At times inexperienced users attempt to use a Runtime license for an IMS server and then don't bother activating that Runtime license. That, of course, won't work.

See Also

Creating a Web Site
IMS Queries
Publishing Multiple Pages
Optimizing Performance
Creating a Web Site

This topic assumes the reader is familiar with Microsoft Internet Information Server (IIS) as well as web programming using HTML and Active Server Pages (ASP). Creating web sites with IIS is a straightforward process supported by a vast array of books and other educational resources. If you are not familiar with the operation of IIS, please get a good book on the subject and learn elementary IIS operation and administration before attempting work with Manifold IMS.


You must be sufficiently familiar with your Windows system to perform elementary tasks such as logging in as Administrator and setting Read security privileges on files and folders for local accounts such as the IUSR_account (or, the NETWORK SERVICE account or ASPNET account if applicable). If you do not know how to do this, please take advantage of the many educational resources available for your Windows version to learn how to do so before commencing work with Manifold IMS.

There are three steps to using Manifold IMS to publish to the Internet:

- Use Manifold to create a project containing the map or other component to be published. The entire rest of this documentation describes the many capabilities of Manifold at your disposal for creating cool maps.
- Create the files required for a map server web page using File - Export - Web Page. This step is described in the Map Server Overview topic.
- Create a web site using the map server web page files. The simplest way is to give users a URL that jumps directly to the created site.

**Very Important:** Before attempting to publish anything using Manifold IMS, please verify correct functionality of your IIS web server. An easy way to do this is to launch Internet Explorer and to load http://localhost/ into the URL address box. This will bring up the default page for your IIS installation or the Windows IIS help if no default page has been created. If you can’t browse a page served by IIS on your machine, stop and get IIS operational before proceeding.

A further check to make sure IIS is serving pages from the directories you think it is using. Perform this check by creating a simple index.htm page and make sure your index.htm page as well as other pages in your web site can be correctly browsed from the machine you intend to use as a server. Please resolve any problems with your IIS installation before attempting to use the Manifold map server.

The Export Web Page dialog creates several files: config.txt, default.asp, default.css and one or more accessory .asp files such as default_admin.asp. It also creates an images directory that contains image files in .png format for toolbar buttons and other controls. We use these files created by the dialog in our web site.

To create a web page using the map server:

1. Verify IIS is running on your machine.
2. In Windows Explorer, Create a directory within C:\InetPub\wwwroot in which the created files are placed. This example assumes we create a directory called "mexico" to publish an exported web page showing a map of Mexico. The location of the new directory is thus C:\InetPub\wwwroot\mexico. This directory must be created at least with Read permissions (which it has by default when a new folder is created in Windows) for the IUSR_account (or, the NETWORK SERVICE account or ASPNET account if applicable).
3. We assume you have already created a Manifold project and have saved it as a .map file. Copy the .map file to the folder in which it will be stored for use by the web site. Using Windows Explorer (not the IIS administration applet) verify the .map file has at least Read permission for the IUSR_account (or, the NETWORK SERVICE account or ASPNET account if applicable).
4. Launch Manifold and open the .map that will be used to create the web page.
5. Open the component that is to be published. Choose File - Export - Web Page.
6. In the Export Web Page dialog's Folder box, browse over to C:\InetPub\wwwroot\mexico. Specify the other boxes and checkboxes as desired and press OK. This creates the web site files within that directory.
7. You can now browse the created map page by entering this URL into the Internet Explorer Address box: http://localhost/mexico/
8. For performance and security reasons it is strongly recommended you create an IIS application for this web page as described below.

9. If for any reason the web page does not work correctly, please verify that the IUSR_ account (or, the NETWORK SERVICE account or ASPNET account if applicable), as at least Read permissions on the .map file involved and in all website folders.

**Note:** once a .map has been used to export a web page, it cannot be moved without altering the .config file used by the web site that specifies the location of the .map that is to be used.

**To create an IIS application:**

1. Open the Internet Information Services manager dialog in Windows [normally accessed from the Start button via Start - Programs - Administrative Tools - Internet Services Manager] and expand the Default Web Site.
2. Within the Default Web Site will be the mexico folder. Right click on the folder and choose Properties.
3. Push Create to create an application. Set execute permissions for Scripts only and High (Isolated) application protection. Press Apply.
4. Press OK to exit the Properties dialog.

When a .map file is in use for one application (map server or Manifold) it cannot be used in another application. This is similar to how most Windows applications work. For information on publishing multiple pages using multiple .map files or using one .map file but publishing different components from the .map file, please see the Publishing Multiple Pages topic.

**About IIS Applications**

An IIS application is any file that is executed within a defined set of directories in a web site. Creating an application designates a particular directory in the web site as the starting point directory (also referred to as the application root). All files and directories under that starting point directory are considered part of that application until another starting point directory is encountered. See the About Applications topic in the IIS documentation.

By placing our Manifold map server files in a particular directory and then designating that directory as an IIS application, we set up IIS so that any processes launched as a result of executing the default.asp file or other files in that directory will be launched as a separate application.

Strictly speaking it is not absolutely necessary to create an IIS application using the directory in which the default.asp file executes. However, this is highly recommended for better performance, reliability and control.

**Running the Web Site**

Web sites created within Microsoft’s Internet Information Server (IIS) can use .asp pages to dynamically create web pages served to browsers. .asp pages provide a sequence of instructions in an ActiveX scripting language that are used to dynamically create web pages on the fly. Based on the contents of the .asp page, IIS will create a stream of HTML content to feed to the requesting browser. Scripting commands embedded in the .asp file can be intermingled with ordinary HTML tags to create exactly the page desired. The default.asp and other .asp pages created by the Export Web Page dialog use javascript for scripting.

The examples above publish the generated web site immediately to the C:\InetPub\wwwroot folder that is the default publication directory for IIS. Browsers will be able to immediately view the web page if given the correct URL. Do not advertise the URL until an application has been created in Internet Information Services manager as noted above and the web site has been tested.

The URL to the new web page follows the usual IIS rules. Suppose we export the web page to C:\InetPub\wwwroot\mexico. Suppose further our web site is associated via DNS with a domain name called manifold.net. Our web site will be viewable at:

http://www.manifold.net/mexico/default.asp
Suppose we have an IIS server running on a Windows machine that is hooked up to the Internet via a cable modem or a DSL link so that it has a fixed IP address. Suppose also we don't have a domain name or DNS but we do have an IP address for the machine such as 192.201.44.168 (even if we have DNS we may not want to hassle with DNS and might prefer to simply used hardwired IP addresses for faster response). In that case we could tell users to browse our web site by referring to:

http://192.201.44.168/mexico/default.asp

Because Microsoft IIS is set up by default to launch a web page called default.asp or index.htm when browsing to a directory, we can also browse our web site using a URL such as

http://www.manifold.net/mexico or

http://192.201.44.168/mexico

The shorter form of the URL (with the default.asp implied) may be more convenient to provide to users.

Windows Server 2003

Very Important: For enhanced security, Microsoft's Windows Server 2003 product comes with Internet Information Server configured so that .asp pages are not enabled. .asp pages must be turned on within IIS for Manifold IMS to function correctly.

Very important: In 32-bit and 64-bit versions Windows Server 2003 and in 64-bit Windows XP the account used for Internet visitors is not the IUSR_ account as with earlier Windows versions. Instead, the NETWORK SERVICE account is used. When ASP .NET is employed the ASPNET account is used.

Windows Server 2003 and 64-bit Windows XP operators should understand the NETWORK SERVICE account is meant in all cases where this documentation discusses the IUSR_ account.

Web pages operating in an ASP.NET environment may require different permission settings than those operating in an ASP environment. For example, the default login used by ASP.NET is ASPNET, while the default login used by ASP is either the IUSR_ account or the NETWORK SERVICE account. When running in ASP .NET, the .map file and all other data being accessed from the ASP.NET environment must be readable by the ASPNET user account.

ASP .NET operators should understand the ASPNET account is meant in all cases where this documentation discusses the IUSR_ account.

IIS 7 and Vista Notes

By default, IIS 7 running in Windows Vista does not allow web applications to create files in the Temp folder. When using IIS 7 in Windows Vista you must allow web applications to create files in the Temp folder. The easiest way to do this is to run the following commands from the Command Prompt with Administrator privileges as follows:

To allow web applications in IIS 7 and Vista to create files in the Temp folder:

1. Click on the Windows Start button and then use the Start Search box.
2. Enter "command" to begin searching for the Command Prompt program. A list of matches will appear.
3. Right click onto the Command Prompt program and choose Run as Administrator. Enter the Administrator password.
4. This launches the Command Prompt window with Administrator privileges. Enter the following two command lines, pressing Enter after each line (enter each line as a single line):

icacls %windir%\serviceprofiles\networkservice\AppData\Local\Temp /grant Users: (CI) (S,WD,AD,X)
We can alter the appearance of a Manifold IMS web page by editing the files that create it:

- Styles used in the default.asp and other .asp pages are kept in the default.css cascading style sheet file. Fonts and colors used may be easily changed by editing the default.css style sheet file.
- Buttons and other images are stored in the images directory. These may be changed or replaced using other images. If the file names are changed (for example, to use .gif images instead of .png images) then the .asp files that use the buttons must be edited to change the names of the images as they are used in those files as well.
- The config.txt file specifies the size of the main map image and the location of the .map file used as well as the standard text used for title, subtitle and copyright lines. We can change the location or name of the .map file to be used by editing the config.txt file.
- We can alter the appearance of the page by editing the .asp pages. The .asp files created by Manifold consists of two types of code: javascript and ordinary HTML commands. The HTML code also may easily be changed to alter the layout, to add additional text to the page and so forth. For example, changes in the table that hosts the copyright text can be made to provide additional lines or legal notices that may be required. Scripting code should not be changed except by experts.
- We can utilize the created code within a frame set page that launches default.asp within one of the frames while using other frames for navigation, explanations, custom title bars or other graphics that provide our web page with a distinctive look and so on. It may be easier for novice webmasters to use frames to provide a custom look while leaving the Manifold .asp files unmodified.
- We can write our own .asp pages that use the WebServer object for a fully custom Manifold IMS application.

**Note:** Changes to config.txt or the .asp files will not take effect until the web application is unloaded. To unload the web application, launch a command prompt window and type iisreset to reset IIS. Changes in the .map file will not be acknowledged until the map server is forced to Reload the .map file, either using the Reload method of the MapServer object programmatically or using the Reo[ld button in the administrative default_admin.asp page.

See the IMS Config.txt Options topic for a list of options that may be used when configuring the config.txt file. Some of these options are not created through the Export Web Page dialog but must be manually entered into the config.txt file.

### Maintenance

The .map file used by the map server cannot be changed while it is in use for publishing the web page except through expert usage of the programmatic interface. The simplest way of changing it is to stop IIS, execute an iisreset command in a Windows Command Prompt window, launch Manifold, change the .map file, close Manifold and then restart IIS. The iisreset command unloads the map server as part of resetting IIS so that Manifold can execute in interactive mode.

It is also possible to change the .map file without restarting IIS. To do this, edit a copy of the .map file and save it under a similar name. For example, if we are using myfile.map with the map server we could edit a copy and save it under the name myfile2.map. Next, change the config.txt file to use myfile2.map instead of myfile.map and then in the Microsoft Internet Information Services manager dialog unload the web page application. Do this by right clicking on the application, choosing Properties and then clicking the Unload button (or, by using iisreset from the command prompt, or, if we have created an Admin page for the web site we could simply navigate to the Admin page and press the Reload button).

**Note:** After unloading a map server application that uses a large .map file the webmaster should browse the site and choose Zoom to Fit with all layers turned on. The first time a large .map file is loaded entirely into RAM the response of the server will be much slower as the entire .map file is loaded. Thereafter, server response will be faster.

For expert usage, IMS opens .map files for both reading and writing. This allows doing sophisticated analysis that involves creating temporary components, such as running queries the text of which is supplied by the user.
Tech Support Tip: The most frequent error users report is accidentally (during maintenance, updates, etc.) changing the security permissions on the .map file in use so that the IUSR_ account (or, the NETWORK SERVICE account or ASPNET account if applicable) no longer has read access permission to the .map file or to the folder in which the .map file resides.

For example, opening the .map file with Manifold can change the permissions on the .map file. If your web site worked fine and then stops working, check this and all other permissions to make sure the IUSR_ account (or, the NETWORK SERVICE account or ASPNET account if applicable) has the necessary access permissions. Saving a .map file with Manifold will preserve file permissions on Windows 2000 and later systems, but this is always something worth checking.

Refreshing Linked Drawings and Tables

Standard templates will refresh linked components as specified in the Refresh linked components option when creating the web page. However, for customized web sites .asp programmers may wish to utilize the properties and methods of the Drawing and Table objects.

The IsLinked method returns True if the component is linked and False otherwise. The LastRefreshed property returns the date and time of the last refresh for linked components and current date and time for non-linked components. The Refresh method refreshes linked components and does nothing for non-linked components. In the case of linked components the method returns True if it has succeeded and False otherwise. Refreshing a table bound to a linked drawing refreshes the drawing and vice-versa. Refreshing a linked table will update formatting based on the table’s columns. The RefreshAfter method does a Refresh if the component has not been refreshed for more than the given number of seconds.

These properties and methods allow programming a website that can force a refresh of a linked drawing or table when it is required by the logic of other parts of the website.

Using Images Linked from Image Servers

At times we may create a Manifold project that includes images linked from image servers. When using such projects within an IMS page, keep in mind that linked images may take a long time to render as image tiles are fetched from the remote image server. For example, an IMS project that contains an image linked from TerraServer won’t render that image on the website until the system fully downloads the entire required portion of the image from TerraServer.

The IMS page including the linked image can be no faster than the image server providing the image. If speed is of the essence, consider downloading the linked image into local storage, perhaps as a compressed image in ECW form.

See the Linked Images topic for links to image server topics.

Server Argument is Optional

When writing custom .asp pages, programmers should note that the Server argument within the WebServer.Create method is optional. Creating a WebServer object without a reference to an ASP Server (or a reference to an ASP-like wrapper for ASP.NET) puts IMS into a reduced functionality mode where it can render pictures, respond to zoom in, zoom out, center and other commands but cannot generate HTML reports. This allows the restricted use of Manifold IMS in ColdFusion and other non-ASP environments that are able to instantiate COM objects but are unable to simulate native ASP objects and are unable to utilize .NET.

Disk Requirements

Manifold IMS creates an image to serve to browsers. Each different view by any browser generates a different image. The image files used are created in the web page’s directory as temporary files in .png format (or other image format specified in the IMS config.txt options). Manifold will cache these images and will delete them after a twenty-minute time out period.

Although the image files served by the map server are small (depending on the size and content usually below 100KB in size), the size of temp files accumulated for a popular site can grow to tens of megabytes. Webmasters should make sure to have adequate free space for temporary files on the disk drives hosting the
web pages in use. In an era of large and inexpensive disk drives it is not normally a problem to be sure to have a few tens of megabytes free space available.

**Other Windows Versions**

The map server has been observed to work using the personal Internet server in Windows 98 and Me; however, these operating systems are not supported for use with the Manifold map server. Performance with 98 or Me (if it works at all) will be greatly reduced as compared to Windows Server 2003, 2000 or XP.

**Security Tips**

Microsoft IIS is a secure and highly reliable web server when correctly configured by the system administrator. Nonetheless, the many options made available within IIS to support the needs of experts can result in security problems if administered improperly. Because an IIS web server connected to Internet can be attacked by any Internet user from anywhere in the world, a few general IIS security measures should be applied to any web server connected to the Internet. These measures are not specific to Manifold IMS; they should be applied to every web server.

We recommend the following elementary steps:

- Use Windows 2003, 2000 or XP. Windows XP configures IIS by default with more security than does Windows 2000. Windows Server 2003 has the most default security.
- Use NTFS file systems.
- Do not allow anyone to work interactively, to read mail or to browse web sites on the machine that hosts your web site. If you must use the same machine interactively, make sure that user logins you employ do not have Administrator privileges. If a user with Administrator privileges just once opens a malevolent email attachment the web server itself could be infected with a wide variety of "backdoor" viruses or Trojan Horse attacks. Never read email or browse the web when logged in as Administrator.
- Install the latest Service Pack for Windows. Service packs include many security patches.
- Configure Windows Update to automatically download and install the latest Windows updates.
- Very Important: If you create a default_admin.asp page for your IMS site, make sure to change access permissions on it so that ordinary Internet browsers cannot use it. To do this, in Windows Explorer right click on the default_admin.asp file, choose Properties and choose the Security tab. Remove all entries for allowed user names except Administrator or whatever other login name you wish to allow access to the administration page. Apply the settings and press OK.
- Use an Administrator password that is called something other than "Administrator". Brute-force password cracking attacks have half the problem solved if they can assume there is a login called "Administrator" to attack.
- Use a seriously long (over 16 digits), randomly constructed password to make it difficult for brute-force password cracking assaults to guess your password.
- Install any supplemental security patches published by Microsoft after the last Windows Service Pack. Microsoft now has tools that can scan your system to determine if there are security patches published that have not yet been installed. Visit Microsoft TechNet at [http://www.microsoft.com/technet/default.asp](http://www.microsoft.com/technet/default.asp) and drill down to their Security pages for information on the latest security tools and patches.
- Install anti-virus software. Keep your anti-virus software updated with regular downloads from your vendor.
- Install Microsoft URLScan (a free download from Microsoft TechNet) or similar tool. URLScan is an ISAPI filtering tool for IIS that rejects a wide variety of malformed URL requests before they can hit IIS. It is very easy to use. It was published by Microsoft in 2001 and is an absolutely essential tool to prevent many different types of Denial of Service attacks. No doubt the functions provided by URLScan will continue to be offered by Microsoft either as independent tools or built into IIS. Manifold IMS works perfectly with URLScan.
- Apply Microsoft's IIS Lockdown Tool (a free download from Microsoft TechNet), or its successors, using Advanced Lockdown. Uncheck the box that disables support for Active Server Pages (.asp) and otherwise accept all of the defaults suggested by Advanced Lockdown. Manifold IMS works perfectly with Advanced Lockdown settings so long as Active Server Pages are still enabled. This tool is also very easy to use. Some recent editions of IIS may be shipped by Microsoft in lockdown mode by default.
- Do not run unnecessary services. If your server runs a web site only, do not install or enable additional services such as FTP, NNTP or others. If such services are already installed, use Internet Services Manager to stop them. Do not install any accessory software or Windows components that are not required.
Do not install unnecessary applications on your web server. Every application you install increases the attack surface that an adversary can exploit. For example, don't install things like Microsoft Office on a machine that will only be a web server.

Do not use your web server, ever, for Internet browsing or other applications. Some people press into use as a web server an otherwise underutilized machine that is occasionally used (out of boredom while waiting for some administrative task to complete, etc.) to browse the web or for other purposes. Dedicate a machine to web serving and don't use it for other purposes.

Do not connect your web server on a local area network to other machines unless you really know what you are doing with security. It is safest to keep it isolated and to use a USB drive to exchange files with other machines.

If you use Remote Desktop to work with your server, change the port it uses to some port other than the default. (Don't forget to leave this port open in your firewall.)

Learn to use Advanced TCP/IP settings for the Internet Protocol in Network properties for the network card in use. If your web site is intended only for specific machines (such as those on an Intranet), use IP security to deny access to any unauthorized machines (IP addresses). If you are connected to Internet, use TCP/IP filtering to permit only those TCP ports absolutely required to service your web site (typically, port 80 for HTTP and port 443 for HTTPS). Strict IP filtering will not allow services such as DNS (used if you are browsing from the web server) so a truly secure regime will limit utilization of the system to a web serving only.

Install a firewall such as Windows Firewall and learn to use it. Use a hardware firewall and/or software solution such as Microsoft's Internet Security and Acceleration Server. Hardware firewalls now cost well under $100. Configure your hardware firewall so that only those ports required for serving web pages are enabled, with all other ports disabled.

Install the latest updates for Manifold System from manifold.net.

Get a good book on Windows and IIS security. Read it carefully and apply the recommendations. Surf the web and read the many free resources on Windows and IIS security.

Participate in Manifold-L (email discussion list format) and the Georeference Forum at http://forum.manifold.net for Manifold System users. Ask questions of your more experienced colleagues and as you gain expertise yourself give back to the community by assisting newbies with security questions.

Very important: Although Manifold IMS can work with Microsoft security tools for IIS such as URLScan and the IIS Lockdown Tool, it is possible that other IIS applications installed on your system cannot work with these tools or require modifications to the default settings of these tools. It is critically important to read the documentation for these security tools.

When establishing security policies it is critically important to keep in mind the need for the IUSR_(or, the NETWORK SERVICE account or ASPNET account if applicable) Internet browser account to have read permissions for all files involved in the web site, including the .map file used as well as any database files used for linked drawings or linked tables. If a .mdb file is used, the IUSR_ account must also have read and write permissions for the folder in which the .mdb file is located, so that the map server process can create a lock file as is required when working with .mdb.

The idea of giving the IUSR_ account such permissions may worry novice webmasters. However, giving the IUSR_ account read permission and knowledge of the path to the .map file (which may be discovered by examining the config.txt file if a visitor knows enough about Manifold System) does not necessarily give the IUSR_ the ability to download the .map file. For example, the .map file could be located in a folder outside the InetPub hierarchy and so not be accessible for download. Likewise, if the folder containing any database files used is located outside the InetPub hierarchy these files cannot be downloaded either.

For novices, a workable approach until more can be learned about Windows security and IIS security is to create a folder, such as C:\lmsMaps, to place the .map file and any database files used in that folder and to grant IUSR_ read/write/execute permission on the folder and read permission for all files within the folder. Adjust the config.txt file in the actual web site (located somewhere in the InetPub hierarchy) so that the path to the .map file in use correctly notes the location in C:\lmsMaps.

Very important: In 32-bit and 64-bit versions Windows Server 2003 and in 64-bit Windows XP the account used for Internet visitors is not the IUSR_ account as with earlier Windows versions. Instead, the NETWORK SERVICE account is used. When ASP .NET is employed the ASPNET account is used.

Windows Server 2003 and 64-bit Windows XP operators should understand the NETWORK SERVICE account is meant in all cases where this documentation discusses the IUSR_ account.

Web pages operating in an ASP.NET environment may require different permission settings than those operating in an ASP environment. For example, the default login used by ASP.NET is ASPNET, while the default login used by ASP is either the IUSR_ account or the NETWORK SERVICE account. When running in ASP .NET, the .map
file and all other data being accessed from the ASP.NET environment **must** be readable by the **ASPNET** user account.

ASP.NET operators should understand the **ASPNET** account is meant in all cases where this documentation discusses the **IUSR**. account.

Note that permissions as discussed above are set using Windows Explorer. Right click onto the folder or file, choose **Properties** and then click on the **Security** tab.

**Troubleshooting**

See the troubleshooting topic Problems with the Internet Map Server for detailed checklists of what might be wrong.

The number one problem with IMS reported to tech support is that users neglect to add the **IUSR**. account (or, the **NETWORK SERVICE** account or **ASPNET** account if applicable) with access permissions to the .map file in use.

Using **Windows Explorer** (do not just depend on the IIS management console or other server management console), right click on the .map file, choose **Properties** and verify in the **Security** that the **IUSR**. account for the system has necessary read and execute permissions.

Take time to think through all permissions issues. The Internet access account **must** have permissions to get to all the data that is used in the web site. For example, suppose our IMS project uses a linked image. Because linked images use data from the Data Cache folder (the default), we must also be sure to give the **IUSR**. account (or, the **NETWORK SERVICE** account or **ASPNET** account if applicable) **read** permissions to access all files in that folder.

Do **not** take permissions for granted if an IMS site has problems. Check each folder and every file with Windows Explorer. For example, if you have opened a .map file with Manifold System it could be that the permissions on that .map file no longer include the **IUSR**. account (or, the **NETWORK SERVICE** account or **ASPNET** account if applicable). Saving a .map file with Manifold will preserve file permissions on Windows 2000 and later systems, but this is always something worth checking.

**About ASP.NET**

**ASP.NET** is a new technology introduced by Microsoft in Windows .NET servers. Microsoft ASP.NET is more than just the next generation of Active Server Pages (ASP). It provides a new programming model for creating network applications.

A key aspect of ASP.NET is that while just about any given task can be implemented in either ASP or ASP.NET, ASP.NET is overall easier to use, much faster, much more robust and much more flexible. Another key thing is that although in earlier editions of Manifold System we could use IMS with ASP.NET by creating a mixed ASP / ASP.NET application with the IMS portion handled by ASP, because IMS is now designed to function directly within ASP.NET environments we now can avoid using ASP at all and have our entire web application written in ASP.NET.

Of course, to use ASP.NET we must be running the Microsoft products that support it. Recent IIS versions are installed by default in "lockdown" mode so that not all facilities are available. In particular, ASP.NET is not enabled. Make sure to enable ASP.NET in your IIS installation using IIS Manager if you will create any IMS web sites utilizing ASP.NET.

**Dependent DLLs in ASP.NET**

ASP.NET allows use of dependent DLLs, which if the system is misconfigured will lead to error messages such as "A dynamic link library(DLL) initialization routine failed."

The most likely reason one or more DLLs are not getting initialized when loaded in an ASP.NET environment is that the system can not locate the dependent DLL modules to which it is linked. To work around this, include the path to the Manifold installation folder into the system PATH variable. That is, go to Control Panel - System - Advanced - Environment Variables and append the path for the Manifold installation folder (typically,
C:\Program Files\Manifold System) to the system (not user!) PATH variable. Then restart IIS, or, better yet, reboot the machine. This should get rid of the problem.

**ASPNET Permissions in ASP .NET**

Web pages operating in an ASP.NET environment may require different permission settings than those operating in an ASP environment. For example, the default login used by ASP.NET is ASP.NET, while the default login used by ASP is IUSR_xxx, where xxx is the machine name. The .map file and all other data being accessed from the ASP.NET environment must be readable by the ASP.NET user account. Use Windows Explorer to check the security settings for all files involved in your web site to make sure that ASP.NET can read them or execute them (in the case of executables such as manifold.exe). See the analogous discussion for the IUSR_xxx account.

**Tech Tip:** A .NET script invoked in the context of a web site can access the Document property of the Context object.

**Manifold System Edition Requirements**

You must be running Manifold System Professional Edition or greater to have the capabilities discussed in this topic. All Manifold System editions except Personal Edition include the map server. If you are running Manifold System Personal Edition you will not have the capabilities discussed in this topic.

If an IMS application attempts to use any features or functions that are part of the Business Tools or Surface Tools packages, it will fail if these extensions have not been installed on the IMS server.

The safest way to avoid configuration errors arising from inadequate licensing is to simply install a Universal Edition Runtime license on the web server machine when deploying an IMS application. Universal Edition includes all the extensions and the Universal runtime license is absurdly inexpensive.

**Required Geocoding Tools**

Manifold's street address geocoding capability becomes operational only when the optional Manifold Geocoding Tools package is installed. If you do not have the optional Geocoding Tools package, you will not be able to use street address geocoding functions within Manifold IMS or take advantage of the street address geocoding data provided on the Manifold downloads site for the United States.

Note that Manifold Universal Edition automatically installs the Geocoding Tools package. If you are using Universal Edition, you do not need to perform an extra Geocoding Tools installation as it has already been installed.

**Required Geocoding Data Sources**

In addition to installation of Geocoding Tools, for any street address geocoding capabilities the Manifold installation must have access to a geocoding data source. Once enabled by installation of the Geocoding Tools package, Manifold can use a variety of geocoding data sources, including the Manifold Geocoding Database for US streets provided on the Manifold downloads site, MapPoint North American or European editions, user-provided geocoding data extensions or Manifold Geocoding Servers accessed through the web.

See the Geocoding Data Sources topic for information on these sources as well as information on installing the Manifold Geocoding Database from the Manifold downloads site.

**See Also**

Map Server Overview
IMS Config.txt Options
IMS Queries
Publishing Multiple Pages
Optimizing Performance
IMS Config.txt Options

The config.txt file contains configuration information for the Manifold Internet Map Server (IMS) web pages that it accompanies. It is an ordinary text file that may be edited with Notepad. The config.txt file is automatically generated by the Export Web Page dialog. A typical config.txt file might look like:

```
component = Mexico Drawing 1
copyright = Copyright (C) 2002. All rights reserved.
cx = 500
cy = 400
file = C:\maps\mexico.map
queries =
subtitle = Click near an edge to pan map. Zoom in to see more layers.
title = Mexico
```

Once a website is generated, the config.txt file may be edited to make simple changes. For example, if the .map file used is moved to a different location the file = string can be changed.

Parameters (with typical examples of usage in the left column) may be:

- **antialiasLines** = true
  - Set to true or false to turn on or off antialiasing of lines.

- **antialiasTexts** = true
  - Set to true or false to turn on or off antialiasing of text.

- **clickAccuracy** = 10
  - Specifies the clicking tolerance in pixels when hyperlinks are clicked or the info tool is clicked on an object. The default accuracy is 4 pixels. Adding a line like that to the left will decrease accuracy to 10 pixels, so that a mouse click within ten pixels of the object will be as if the mouse were clicked directly onto the object.

- **Component = MyMap**
  - Name of component within the project to use. For example, if you have a map component named "MyMap" and you want that to be the component displayed by IMS.

- **configurationFolder = \%Manifold\%\Config**
  - The location of the Configuration folder, as given in Tools - Options, File Locations.

- **Copyright = Copyright (c) Acme Corp, 2005**
  - String to use in copyright position in standard templates.

- **country = United States**
  - Default country name for use by the geocoding engine.

- **cx = 800**
  - Width of the generated image in pixels.

- **cy = 600**
  - Height of the generated image in pixels.

- **dataCacheFolder = \%My Documents\%**
  - The location of the Data Cache folder, as given in Tools - Options, File Locations.

- **datumGridsFolder = \%Manifold\%\Grids**
  - The location of the Datum Grids folder, as given in Tools - Options, File Locations.

- **File = C:\data.map**
  - Absolute path to file to use.

- **geocodingDatabaseFolder = \%Manifold\%\GCDB**
  - The location of the Geocoding Database folder, as given in Tools - Options, File Locations.

- **geocodingExtensionsFolder = \%Manifold\%\GCDBX**
  - The location of the Geocoding Extensions folder, as given in Tools - Options, File Locations.

- **geocodingServers = Geocoder.us, mygcdbserver.org**
  - A list of Manifold Geocoding Servers to use by the geocoding engine, separated by commas.

- **hyperlinks = true**
  - Enable hyperlinks.
hyperlinksNew = false

If true, launch hyperlinks in new browser.

imageDesc = true

Embeds text information into the generated .png image.

logfile = c:\\mapserver.log

Specifies the path to the logfile, if used. By default, a logfile is not used. If used, a logfile will log useful parameters, for example, the time required to render the legend, if a legend is used.

logo = true

Available for Enterprise Edition users. If false, suppresses "Powered by manifold.net" logo.

mrSidDecodeUtilityPath = MrSidDecode.exe

The path to the MrSID Decode utility, as given in Tools - Options, File Locations.

mrSidInfoUtilityPath = MrSidInfo.exe

The path to the MrSID Info utility, as given in Tools - Options, File Locations.

ogcWmsCaps = C:\Inetpub\Wwwroot\filename

Specifies the path to the custom server capabilities document, if used, for OGC WMS.

preferMapPointToGeocodingDatabase = false

Set to true or false. If true, gives MapPoint priority over Manifold Geocoding Database as a geocoding data source for the geocoding engine.

Proxy = true

Proxy server address.

proxyAddress = http://netproxy:8080

Password used to login onto the proxy server.

proxyPassword = 123passwd

User name used to login onto the proxy server.

proxyUser = rootuser

Set rendering format. Supported rendering formats are gif, jpeg and png.

renderFormat = png

renderOptions = dither

Set rendering options. Supported rendering options are dither (supported for gif rendering format) and interface (supported for png rendering format).

renderQuality = 100

Set rendering quality in the range of 1 to 100, with 100 being the maximum quality.

queries = "Query 1", "Query 2", 52

Specifies queries used by IMS, using their names (such as "Query 1") or IDs (such as 52).

queryLimit = 100

Specifies the maximum number of records returned to the user by a query. The default value is 200.

refreshLinks = 60

Time interval over which linked components used in the website will be refreshed, in minutes. A value of 0 will refresh links on each browser access (can help debug server-side logic, not recommended for servers running in production mode due to heavy performance requirements).

refreshLinksOnOpen = true

Set to true or false. When true (the default), automatically refreshes linked components in opened MAP files. When false, does not automatically refresh linked components to save time.

scaleNumeric = true

Set to true or false. When true, causes scale (if used) to be reported as the horizontal extent of the visible region shown in the map server image. When false, scale is reported in absolute scale form.

scaleEnglish = true

Set to true or false. When true, turns on the use of English measurement units when reporting scale in horizontal extent form, otherwise scale will be reported in metric units.

selectionColor = 255:0:0

Specify color in either hexadecimal or decimal form to use
for selection color within IMS.

Hexadecimal colors are in "#rrggbb" form where rr, gg and bb are pairs of hexadecimal digits giving a hexadecimal number corresponding to the decimal number range from 0 to 255 for Red (rr), Green (gg) or Blue (bb). Leading zeros are significant, so that RGB values of 3, 5 and 6 are coded as 030506 in hexadecimal numbers. For example, selectionColor = #54BBB3 will set selection color to a light green-blue color instead of the default red selection color.

Decimal colors may be in two forms: either r:g:b form where r, g and b are decimal integer numbers in the range from 0 to 255 or as a single decimal number that is the decimal equivalent of a #rrggbb hexadecimal number.

selectionStyle = dashes

Controls the selection style used to display selections. Possible option values are: border, denseDots, mediumDots, sparseDots, and dashes. Case does not matter. For example, selectionStyle = border will change the selection style used to a border outline for areas.

smoothLargeVectorObjects = true

Set to true or false to turn smoothing of large vector objects on and off. Turning smoothing on increases performance at a slight cost in rendering quality.

Subtitle = Use the toolbar buttons to switch between zoom and pan modes.

String to use in subtitle position in standard templates.

Title = My Map

String to use in title position in standard templates.

useGeocodingDatabase = true

Set to true or false. Toggles the use of the Manifold Geocoding Database as a geocoding data source by the geocoding engine.

useGeocodingExtensions = true

Set to true or false. Toggles the use of geocoding data extensions as a geocoding data source by the geocoding engine.

useMapPointEurope = true

Set to true or false. Toggles the use of MapPoint Europe as a geocoding data source by the geocoding engine.

useMapPointNorthAmerica = true

Set to true or false. Toggles the use of MapPoint North America as a geocoding data source by the geocoding engine.

For example, we can edit the above config.txt file as follows:

```
component = Mexico Drawing 1
copyright = Copyright (C) 2005. All rights reserved.
cx = 500
cy = 400
file = C:\maps\mexico.map
queries = subtitle = Click near an edge to pan map. Zoom in to see more layers.
title = Mexico
scaleNumeric = true
scaleEnglish = false
```

This will cause the scale to be reported as the number of kilometers spanning the horizontal extent of the displayed map server image.
A useful way of seeing what a config.txt file looks like is to create a web site, checking the Save proxy connection and file locations in the config file option checkbox, and then open the config.txt file to see what Manifold has created.

**Manifold System Edition Requirements**

You must be running Manifold System Professional Edition or greater to have the capabilities discussed in this topic. All Manifold System editions except Personal Edition include the map server. If you are running Manifold System Personal Edition you will not have the capabilities discussed in this topic.

**Required Geocoding Tools**

Manifold's street address geocoding capability becomes operational only when the optional Manifold Geocoding Tools package is installed. If you do not have the optional Geocoding Tools package, you will not be able to use street address geocoding functions within Manifold IMS or take advantage of the street address geocoding data provided on the Manifold downloads site for the United States.

Note that Manifold Universal Edition automatically installs the Geocoding Tools package. If you are using Universal Edition, you do not need to perform an extra Geocoding Tools installation as it has already been installed.

**Required Geocoding Data Sources**

In addition to installation of Geocoding Tools, for any street address geocoding capabilities the Manifold installation must have access to a geocoding data source. Once enabled by installation of the Geocoding Tools package, Manifold can use a variety of geocoding data sources, including the Manifold Geocoding Database for US streets provided on the Manifold downloads site, MapPoint North American or European editions, user-provided geocoding data extensions or Manifold Geocoding Servers accessed through the web.

See the Geocoding Data Sources topic for information on these sources as well as information on installing the Manifold Geocoding Database from the Manifold downloads site.

**See Also**

- Geocoding Data Sources
- Geocoding Tools
- Map Server Overview
- Creating a Web Site
- IMS Queries
- Publishing Multiple Pages
- Optimizing Performance
IMS Queries

Web pages created with the Manifold map server can utilize queries that operate on the component being published. If any queries exist in the project they will be made available in the Queries pane of the Export Web Page dialog.

For example, if our project contains three queries as seen above, launching File - Export - Web Page will show them in the Queries pane as seen below:

Checking a query's checkbox will cause it to appear in the Queries pane of the generated web page. If we check all boxes in the Export Web Page dialog all three queries will appear in the web page:

Two of the queries are parameter queries - they allow user entry of a parameter and so are equipped with an edit box allowing user entry. The third query takes no parameter and so has no edit box.

Entering a value into a query edit box and pressing Query will launch the query.
The result is a table that lists the output of the query. Tables will appear either in a new browser window or in a frame in the same browser window depending on which template was used to create the web page. Any URL columns will have hyperlinks created for their entries in the table.

Optional templates used within the Export Web Page dialog will create different pages. The configuration seen in the browser window depends on choices made in the dialog. The illustrations shown in this topic use the Standard template, which is designed for use with newer browsers. The Compatible with 4.x browsers template uses a simpler style.

Tables in sites created with the Compatible template will be displayed in the same browser window, thus requiring a back command in the browser to go back to the map page.

Captions

The names of the queries are used for the text strings that appear in the Queries pane in the web page. The captions for the edit boxes are simply the name of the parameter variable used in the parameter query.

For example, the States with population query shown above contains the following SQL:

```sql
PARAMETERS exceeding LONG;
SELECT NAME, POP1990 FROM [States Drawing] WHERE POP1990 > exceeding;
```

The parameter variable exceeding is accepts user entry and is used within the query to process the SQL statements.

Manifold will use the name of the query, States with population, together with the name of the parameter variable, exceeding, to construct the captions for the query. When creating a Manifold project for publication to the web it therefore is important to think ahead when naming queries and parameter variables so that the resulting captions are reasonably well self-documenting.

Rules

When using queries in web pages the following rules must be observed:

- All queries in the project will be listed in the Queries pane in the Export Web Page dialog, even those that cannot legally be used within a map server site. It is the Manifold operator's responsibility to know the contents of the queries and to choose acceptable queries.
- Only SELECT queries may be chosen for use in the generated site.
- The Manifold operator is responsible for selecting queries that operate within the component being published. For example, if a project contains a query that reference a drawing that is not part of a map that was published, the query should not be chosen for use in the site.
- Queries output a maximum of 200 records to a table on the web page.
Launching a new query will eliminate the results of the previous query.

To avoid errors, it is strongly recommended that projects used with the map server contain no unnecessary or unusable components.

**Examples**

The **States matching** query seen in the illustration at the beginning of this topic contains:

```sql
PARAMETERS pattern TEXT;
SELECT NAME FROM [States Drawing] WHERE NAME LIKEX pattern;
```

It uses the **LIKEX** operator to find the names of states matching the pattern provided by the user.

The **10 longest roads** query contains:

```sql
SELECT TOP 10 LENGTH, TYPE, ROUTE FROM [Roads Drawing]
ORDER BY LENGTH;
```

There is no parameter used in this query. It simply always selects the ten longest roads based on the contents of the **Length** field.

**Panning to Selected Objects / Suppression of ID Field**

Objects are not selected by queries in Manifold unless the **ID** system field is selected by the query. For example, if we have a drawing of provinces that each has a name, the SQL fragment ...  

```sql
SELECT Name from [Provinces Table] WHERE...
```

... will create a table, but it will not select the objects associated with the records that appear in that table. In contrast, the fragment...

```sql
SELECT ID, Name from [Provinces Table] WHERE...
```

... will select the objects as well as creating the table.

If the first SQL fragment is part of a query that is used in a Manifold IMS web site the output of the query will be a table, but the IMS map window will not be panned or zoomed. If the second IMS fragment is used, a table will be created, the objects will be selected, and the IMS map window will pan to the center of the selected set of objects and will zoom to fit the selection.

Since we normally do not want to show internal system fields such as the **ID** field to people browsing our web site, the **ID** field will not be displayed as part of a table created by an IMS query.

**Geocoding and Queries**

The Manifold geocoder is accessed in IMS applications by using queries utilizing the geocoding SQL extensions. Manifold SQL includes geocoding SQL extensions that operate with Manifold’s geocoding engine to perform spatial operations based upon an address string or zip code. Geocoding queries will not function unless either Manifold's US streets geocoding database is installed or Microsoft’s MapPoint program is installed. If MapPoint is used, performance will be lower but coverage will extend to all countries covered by the MapPoint editions (North America and/or Europe) installed. See the Geocoding with MapPoint topic if MapPoint is to be used. Note that Internet access permissions must be adjusted (see below) if MapPoint is called from IMS.

Geocoding extensions will not work with Manifold IMS using the Manifold US streets geocoding database unless the geocoding database is installed within the Manifold application installation folder (normally, C:\Program Files\Manifold System). Therefore, the US streets geocoding database should be installed in the Manifold application installation folder on machines on which Manifold IMS operates. Geocoding queries using MapPoint
will work so long as MapPoint is correctly installed on the server system regardless of the installation directory used.

Drawings must be projected for geocoding extensions to function correctly.

**Boolean CloseToAddress(Number ID, String Address, Number Distance, [String Unit])**

Given an object ID, an address string, a distance and an optional distance unit determine if the object lies within the specified distance of the address.

**Boolean CloseToZip(Number ID, String Zip, Number Distance, [String Unit])**

Given an object ID, a ZIP code string, a distance and an optional distance unit determine if the object lies within the specified distance of the zip code centroid.

**Number DistanceToAddress(Number ID, String Address, [String Unit])**

Given an object ID, an address string, and an optional distance unit computes the distance between the object and the address.

**Number DistanceToZip(Number ID, String Zip, [String Unit])**

Given an object ID, a ZIP code string, and an optional distance unit computes the distance between the object and the zip code centroid.

**Notes on usage:**

- If an object is a line or an area the object's centroid is used for distance calculations.
- If an address string produces more than one match, the system automatically selects the closest of the building-level matches (possibly with an "unknown street name, possible misspelling" error).
- If an address string produces no building-level matches, `CloseToAddress` returns `False` and `DistanceToAddress` returns `-1`.
- If a zip code string is invalid (no matches in the geocoding database) `CloseToZip` returns `False` and `DistanceToZip` returns `-1`.
- If the optional distance unit is omitted, the system will use the native measurement unit of the drawing or meters if the drawing is not projected.
- Distances are great circle distances computed over a WGS84 ellipsoid and are accurate to 1 meter.
- The geocoding functions cache returned geocoding data between subsequent calls.
- Functions can be used from IMS as long as the geocoding database is located within the Manifold application installation folder (usually `C:\Program Files\Manifold System`).

**Geocoding Function Examples**

```
SELECT * FROM Dealers
WHERE CloseToAddress(ID, "330 Lytton Ave, Palo Alto, CA, 94301", 10, "mi")
```

```
SELECT * FROM Dealers
WHERE DistanceToAddress(ID, "330 Lytton Ave, Palo Alto, CA, 94301", "mi") <= 10
```

```
SELECT * FROM Dealers
WHERE CloseToZip(ID, "94301", 10, "mi")
```

```
SELECT * FROM Dealers
WHERE DistanceToZip(ID, "94301", "mi") <= 10
```
All four examples have a similar function. The first query selects all objects in Dealers that are within 10 miles of the given address using the CloseToAddress function, while the second example performs the same task using the DistanceToAddress function. The third and fourth examples perform the same functions using the 94301 ZIP code.

To keep the user interface simple and to avoid the complication of dealing with possible user errors when entering address information into forms, many web applications with IMS will use the CloseToZip or DistanceToZip functions since these require the user to merely enter the ZIP code correctly. For many applications, such as locating a dealer, finding the closest objects to the ZIP centroid provides acceptable accuracy.

See the Units topic for a list of unit abbreviations that may be used to specify optional distance units.

MapPoint and IMS

MapPoint geocoding will normally not work from IMS unless we map anonymous Internet connections to a user account that has more enhanced permissions than the default Internet access IUSR_xxx account.

To enhance permissions for the Internet access account:

1. Create a regular user account with default permissions that is a member of the Users group.
2. Open Control Panel - Administrative Tools - Internet Information Services (or the equivalent dialog in your version of Windows).
3. Right click the folder that contains the published website, select Properties, switch to Directory Security and click Edit under Anonymous access and authentication control.
4. Set the user account used for anonymous access to the account created in step 1 above.

See the Geocoding with MapPoint topic for use of MapPoint as an auxiliary geocoder.

Tech Tip

When developing on one machine and publishing on a different machine, make sure the machine that is actually running the map server is a fully up-to-date Windows machine with all required updates in place just like the machine that was used to create and test the project.

For example, Manifold System requires that Microsoft Internet Explorer 5.5 or later (preferably the most recent IE version) has been installed to make sure that all Windows components, including scripting components, have been updated. Occasionally, server machines used as IIS hosts might not have IE installed or might not have the latest version of IE installed because the server is not used interactively to browse the web.

In such cases a Manifold project and web page might work fine on the development machine but then not work correctly (queries fail, etc.) when installed on the server. If this happens, check the server machine to make sure Windows is fully updated as required by your Manifold System installation notes.

Also make sure to verify that the web server machine has all required Manifold licensing and accessories installed. For example, if an IMS web site uses a project that depends upon Enterprise Edition features the web server machine must also have a license for Enterprise Edition and the project must be correctly configured in the web server so it can connect to the Enterprise server it uses. Likewise, if geocoding queries are used the web server machine must have the Manifold Geocoding Data product installed and be licensed for its use.

Troubleshooting

See the troubleshooting topic Problems with the Internet Map Server for detailed checklists of what might be wrong.

The number one problem with IMS reported to tech support is that users neglected to add the IUSR_account the IUSR_account (or, the NETWORK SERVICE account or ASPNET account if applicable) with access permissions to the .map file in use. Using Windows Explorer (do not just depend on the IIS management console or other server management console), right click on the .map file, choose Properties and verify in the Security that the IUSR_account for the system has necessary read and execute permissions.
Take time to think through all permissions issues. The Internet access account must have permissions to get to all the data that is used in the web site. For example, suppose our IMS project uses a linked image. Because linked images use data from the Data Cache folder (the default), we must also be sure to give the IUSR_ account (or, the NETWORK SERVICE account or ASPNET account if applicable) read permissions to access all files in that folder.

Windows security and access permissions can be confusing, but it should be remembered that these issues are Windows issues and not Manifold issues. For example, if an IMS project includes calls to DBMS tables resident in an external DBMS, such as SQL Server, the application will operate using the security context of the ASP or ASPX page comprising the IMS application. Resolving connection issues (such as a failed login to the SQL Server) must be resolved with an informed understanding of Windows integrated authentication.

When troubleshooting such problems, the first objective should be to make sure you thoroughly understand all Windows security mechanisms, especially those involved with IIS applications.

**Note**

You must be running Manifold System Professional Edition or greater to have the capabilities discussed in this topic. All Manifold editions except Personal Edition include the map server. If you are running Manifold System Personal Edition you will not have the capabilities discussed in this topic.

**See Also**

Queries
Map Server Overview
Creating a Web Site
Publishing Multiple Pages
Optimizing Performance
Spatial Extensions
Geocoding Extensions
Street Address Geocoding
Geocoding with MapPoint
Publishing Multiple Pages

Manifold IMS can publish multiple pages at once. There are two main ways to publish multiple pages:

- Publish several different pages using several different .map files.
- Publish several different pages using several different components within the same .map file.

**Multiple Pages using Multiple .map Files**

To publish multiple map files, create a separate directory for each web page to be created. Place the config.txt, default.asp and default.css files created for that page by File - Export - Web Page into the directory along with an image subdirectory containing the standard toolbar button images.

For example, if we published a map of Mexico we might place that in C:\inetpub\wwwroot\mexico\ and if we also published a world map we might place that in C:\inetpub\wwwroot\world\.

Create a separate IIS application for each directory containing the files for that page.

For example, in the screenshot above we see an Internet Services Manager of a web site with several IIS applications. Three of them (manifold, mexico and texas) are Manifold IMS applications created using three different .map files. Each of the directories contains the config.txt, default.asp and default.css files created for that web page by the File - Export - Web Page dialog. Each of the application directories also contains a directory called images with the standard image files for toolbar buttons and controls that is also created by the Export Web Page dialog.

It may seem wasteful to duplicate the images directory for every application, but the images used are of negligible size. Duplicating the images directory for each page served by the map server simplifies the default.asp code and also allows easy customization of the appearance of toolbar buttons for each page if so desired.

**Multiple Pages using the Same .map File**
Different components from the same .map file can be used with different map server pages. However, all map server pages must be part of the same IIS application. To do this, create one directory with several subdirectories for each page. Create the IIS application on the parent directory.

Suppose, for example, we have created a usa.map file containing several drawings showing different data sets for the United States. One web page will show a map that contains several layers of interest to education. Another will show a map containing income layers and a third will show population layers.

We might create directories as follows:

- C:\InetPub\wwwroot\usa with subdirectories
- C:\InetPub\wwwroot\usa\education
- C:\InetPub\wwwroot\usa\income and
- C:\InetPub\wwwroot\usa\population

The education directory would contain the config.txt, default.asp and default.css files for the education map. The income directory would contain the config.txt, default.asp and default.css files for the income map, and the population directory would contain the files for the population map. The config.txt files for all three pages would use usa.map as the originating map file; however, the component used would be different.

We would create the IIS application on C:\InetPub\wwwroot\usa. All processes launched, including those for the default.asp files in education, income and population would thus be part of the same application and could use the same .map file.

The illustration above shows the application and directory structure within Internet Service Manager.

**Troubleshooting**
See the troubleshooting topic Problems with the Internet Map Server for detailed checklists of what might be wrong.

The number one problem with IMS reported to tech support is that users neglected to add the IUSR_ account to the map file in use. Using Windows Explorer (do not just depend on the IIS management console or other server management console), right-click on the .map file, choose Properties and verify in the Security tab that the IUSR_ account for the system has necessary read and execute permissions.

Take time to think through all permissions issues. The Internet access account must have permissions to get to all the data that is used in the web site. For example, suppose our IMS project uses a linked image. Because linked images use data from the Data Cache folder (the default), we must also be sure to give the IUSR_ account (or, the NETWORK SERVICE account or ASPNET account if applicable) read permissions to access all files in that folder.

Note

You must be running Manifold System Professional Edition or greater to have the capabilities discussed in this topic. All Manifold editions except Personal Edition include the map server. If you are running Manifold System Personal Edition you will not have the capabilities discussed in this topic.

See Also

Map Server Overview
Creating a Web Site
IMS Queries
Optimizing Performance
Optimizing Performance

For a given level of hardware speed and capacity, performance for a Manifold IMS map server page depends upon Windows system settings (including IIS settings) as well as the configuration of the map server and the .map file it uses.

The map server uses Manifold to generate images served via the web page. The basic performance of the map server will be similar to that of Manifold during interactive usage and is subject to the same optimization tips suggested for Manifold in the Performance Tips topic. Make sure you follow all the performance tips listed in that topic.

Additional performance tips of special interest to IMS:

**Manifold IMS Performance Tips**

- **Use 64-bit Windows and 64-bit Manifold** - 64-bit operation is significantly faster than 32-bit operation in IIS applications because larger amounts of memory can be more effectively used.

- **Export smaller sized web pages** - Generating a map server image of 400 x 300 requires less computation than an image of 800 x 600. Larger views will usually include more objects and so require greater display computation. In addition, serving larger images means moving larger amounts of pixels into the created image and then transmitting those larger numbers of pixels down the Internet pipes connecting the server to the user's browser.

- **Avoid use of very large .map files** - The larger and more complex the component the slower will be the response of the map server. Just as with interactive use of Manifold, performance of the map server will improve as the user zooms in so that fewer objects participate in the display.

- **Avoid unnecessary use of images and surfaces** - Images and surfaces can be slower than drawings for the information they provide. The Manifold map server will happily serve images and surfaces but these should not be used as casual substitutions for drawings when a drawing will do as well and will provide higher performance.

- **Use Compressed Images** - Using a linked image with an ECW or JPEG 2000 file is vastly faster than ordinary images, a difference of virtually instantaneous zooming and panning as opposed to possibly minutes for very large images.

- **Use Oracle Spatial or a spatial DBMS for Images** - Images linked from an Oracle Spatial server or from a spatial DBMS managed by Manifold will be extremely fast at virtually any size. Linking images from Oracle requires Enterprise Edition, but any Manifold edition can link images from spatial DBMS managed by Manifold.

- **Install Ample RAM** - As components in the .map file are used, they are read off disk and brought into RAM. Once a component is in RAM, it stays there unless the RAM must be freed for some other purpose. The map server can work faster if the components it needs are already in RAM. As the parts of the .map that are served are brought into RAM and stay there (as a result of normal Windows RAM caching) access speed will increase. This only works, of course, if there is enough RAM so that the cached .map data is not flushed from RAM because some other Windows application needs the RAM. RAM is cheap: install many gigabytes if you can do so. Choosing a server motherboard that can host 16 or 32 GB of RAM is much less expensive than having to use more than one server machine.

- **Run Multi-core Processors** - Almost anything that can be done to avoid having to scale out to multiple server machines is worth the cost and effort. Choose a motherboard with two or more processor sockets and then install quad-core processors to get eight or more processor cores in action. If you have multiple processor cores don't forget to configure multiple worker processes. For example, if you have a quad core processor and want to max out all four cores, configure your web application to have four worker processes.

- **Eliminate initial load lag** - The first time a very large map is served the map server will be slow to respond while the .map is opened and loaded for the first time. After creating a site the webmaster should browse the site and choose Zoom to Fit with all layers turned on. This will load the entire .map file into RAM. Thereafter, server response will be faster.

- **Match component projections to maps** - If a map is served, all layers in that map should have exactly the same native projection as used by the map. This will eliminate any need to re-project on the fly. This is especially important for images and surfaces.

- **Delete unused components from the .map file** - If the .map file contains many components other than those published in the web page the map server will respond slightly more slowly than if the .map file contained only those components being published. Delete any unused components from the .map file. This is normally a very minor impediment to maximum performance. However, if the .map file contains very large components that are not used the performance difference may be noticeable.

- **Delete unused data from components** - If the web site does not enable the Find tool or use queries there is no point in publishing a drawing that contains data attributes. Delete the columns in the drawing's table. When publishing maps that contain more than one drawing the Find tool will list all
fields in all drawings in the map. Delete all fields that you do not want users to be able to employ in Find. This will reduce the size of the .map file.

- **Avoid use of unnecessarily detailed data** - A web site that provides overall, national data should not use extremely detailed high-resolution drawings. Use Normalize Topology with appropriate Location Precision settings to simplify drawings. This is often a key step to increasing the speed of redisplay, pan, zoom, etc.

- **Use Zoom Ranges to present the level of detail required** - Suppose we would like to show a national map for user orientation when zoomed out but we would also like to show very detailed shorelines when users zoom in. Create a map using two drawings: a highly detailed drawing plus a second, low resolution drawing created from the first by generalizing with Normalize Topology to a lower Location Precision value. Assign zoom ranges to the drawings so that the low-resolution drawing is the only drawing displayed when zoomed out and the high-resolution drawing is the only drawing displayed when zoomed in. When users browse the map in a zoomed out view the display will be fast because a fewer number of coordinates need be used to display the low-resolution drawing. When users are zoomed in the high-resolution drawing will become visible; however, only a portion of the high-resolution drawing will be seen in a zoomed in view so the display will still be fast.

- **Store the .map file used on a fast, local hard disk** - When the Manifold map server operates it accesses the .map file that was used with File - Export - Web Page to create the web page. If the .map file is located on a slow hard disk or on a different machine that must be accessed via a local area network then performance will not be as fast as if the .map file was immediately available on a local, fast hard disk.

- **Avoid using intrinsic fields in queries**. For example, if the length of a road line must be used, do not compute the length on the fly via the Length (I) intrinsic field. Instead, create a new column called Length and copy the contents of the Length (I) intrinsic field into this new column using the transform toolbar Copy command. When an intrinsic field is used in a query a small computation must be performed for every record for every query. If the intrinsic field is copied to a regular column the computation is done only once. The query can then use the regular column for faster operation.

- **Avoid unnecessary refreshes**. Depending on the technology employed, the speed of the communications channel and the amount of data involved each refresh can impact performance. There is no need to refresh every minute if data can only change significantly once per hour.

- **Avoid slow linked components**. While linked components are the perfect solution in many applications, at times it does not make sense to use them. At other times it makes sense to use linked components but only if used correctly, such as by an informed, expert choice of which type of linked component to use. For example, it may be tempting if we need a background image for a web site to simply link in an image from an OGC WMS server to which we have access. However, thereafter whenever we refresh links our web site will be at the mercy of whatever performance is provided by the OGC WMS server. It would be faster and better (assuming the background image does not change, as they rarely do) to convert that image into a local image such as an ECW image. See the discussion of refresh issues in the Map Server Overview topic.

- **Use EPSG projections for OGC WMS**. If the coordinate system (projection) used by a served component is not supported by the EPSG schema used by the OGC WMS community, Manifold will have to re-project the component on the fly into a Projection (I) intrinsic field. Instead, create a new column called Projection (I) and copy the contents of the Projection (I) intrinsic field into this new column using the transform toolbar Copy command. When an intrinsic field is used in a query a small computation must be performed for every record for every query. If the intrinsic field is copied to a regular column the computation is done only once. The query can then use the regular column for faster operation.

- **Use fast database technology**. Manifold works with virtually all DBMS technologies, which is great for flexibility but also gives us the ability to choose either very slow or very fast technologies. For example, some ADO .NET connections can be 600 times faster than an ODBC connection to the very same data source. Since many Manifold IMS web sites will include the use of some database technology (for example, if linked tables are used or linked drawings) it is important that we choose such technology wisely. In the case of linked tables if the data originates in a slow technology such as an Excel .xls file it will usually make sense to upgrade to a very fast technology, such as a SQL Server table accessed via ADO .NET or an Oracle table that can be accessed via OCI. As a rule of thumb, the fastest connections for storing attributes are to use ADO .NET with SQL Server or to use Oracle (which automatically uses OCI). The fastest geometry storage for large applications is to use a spatial DBMS such as SQL Server 2008 spatial, the Manifold spatial extender with SQL Server 2005, IBM DB2 with Spatial Extender, PostgreSQL/PostGIS or Oracle (taking advantage of Oracle Spatial or Locator services in Oracle XE or in standard Oracle).

- **Very Important**: If you create a default_admin.asp page for your IMS site, make sure to change access permissions on it so that ordinary Internet browsers cannot use it. To do this, in Windows Explorer right click on the default_admin.asp file, choose Properties and choose the Security tab. Remove all entries for allowed user names except Administrator or whatever other login name you wish to allow access to the administration page. Apply the settings and press OK. Neglecting this important step will not create a security breach in that hackers will be able to steal your data, but it will allow a malevolent user to effectively shut down your map server by repeatedly causing it to Reload .map files.
- Create an application and consider using High (isolated). - Create an application on the directory in which the web page is located and run it with high isolation. This runs the map server process as an isolated process outside of the IIS process and separate from any other applications running on IIS. This helps performance slightly (albeit at the cost of creating temporary files for every user), and also isolates the application so any errors in the scripts or map server application won't bring down IIS or any other application. For performance reasons, don't run more than ten IIS applications as High (isolated). To run additional map server applications beyond ten, if the map server application is well debugged (that is, the scripts used in default.asp and other files are well debugged), use Medium (pooled) for the additional applications. Medium pooling might also be a good idea to reduce the creation of per-user temp files (see comments below).

- Don't enable session state - Within the IIS manager, in Properties - Configuration - App Options uncheck Enable session state. The Manifold map server does not use sessions so this can be unchecked if desired to eliminate the overhead of maintaining session state. In the same dialog it is a good idea to also uncheck Enable parent paths if desired. This has no impact on performance but it does improve security. Unchecking this box means the application can not refer to anything in higher directories. The map server application itself only refers to files within its own directory or subdirectories such as the images folder.

- Enable object pooling - If there are more than about 10 simultaneous users, enable object pooling. This is done in the Control Panel - Administrative Tools - Component Services manager. Drill down into COM+ Applications and find your map server application. Drill down into Components for that application and right click on the component there and choose Properties. Object pooling settings are on the Activation tab. Check the Enable object pooling box. The pooling parameters may be left on default settings.

**System Configuration Performance Tips**

- Store the .map file used on a fast, local hard disk - When the Manifold map server operates it accesses the .map file that was used with File - Export - Web Page to create the web page. If the .map file is located on a slow hard disk or on a different machine that must be accessed via a local area network then performance will not be as fast as if the .map file was immediately available on a local, fast hard disk.

- Configure temp folders on a separate, fast hard disk - Putting temp folders on a separate hard disk allows "overlapping head seeks" so that the disk drivers can command a head movement on the disk(s) holding the temp folders at the same time heads are moving on whatever disk(s) host the rest of what is going on in Windows. This is a fine point, but not one to be overlooked if you are trying to wring maximum speed out of your system. Head motions are the slowest thing in the system so we don't want to waiting around for one head motion to finish before we get going on the next.

- Use Large Disks and RAID - Large disks are faster than smaller disks. Use modern interfaces such as SATA and configure several hard disks in a striped RAID array for speed.

- Install lots and lots of RAM - Installing plenty of dynamic RAM is often the most efficient way to enable a server to serve more users faster. RAM is now so inexpensive (under $60 for a gigabyte) that it makes sense even for a hobbyist to install two or three gigabytes of RAM. If you are running an older motherboard that can hold only one and a half gigabytes, fill it up with the maximum amount of RAM right away it even if you don't feel you need it because eventually memory for older motherboards will be difficult to get.

- Use a fast processor - A faster processor will be able to serve more users. Upgrade to a gigahertz class processor.

- Use a Quad Core 64-Bit Processor - Quad-core processors do not cost significantly more than dual-core or single-core processors. Use 64-bit hardware with 64-bit Windows and 64-bit Manifold. Using 32-bit software and hardware in a web server application is a terrible waste.

- Use multiple processors or multi-core processors - Each new release of Manifold adds more functions that are multi-threaded. For example, Manifold will use more than one thread to render image libraries if more than one processor or processor core is available on the computer system. Therefore, image libraries will render faster on multiprocessor or multi-core processor systems such as those using two sockets to host a quad-core processor in each socket.

- Use Windows Server 2003 or XP - Windows Server 2003 provides the best performance. Windows XP provides slightly better performance than Windows 2000. Do not use older Windows releases. Vista is cool, but at the present writing it is not as fast as Windows Server 2003 or XP x64.

- Deploy a server farm - Like any web application, given enough users even a fast IMS application can be overwhelmed if only a single machine is used. Novice webmasters are sometimes surprised to learn that large sites serving many visitors like Google, airline reservation sites or mapping sites don't just run on a single computer but are hosted by hundreds or even thousands of computers configured in a server farm. If we want to compete with a huge site hosting millions of visitors per hour, no matter how fast Manifold may be we too may need to deploy a warehouse full of servers.

- Plan to use NVIDIA CUDA - Try to choose a server motherboard that can support at least one NVIDIA CUDA-capable GPU. Although this technology is just getting started, each new release of Manifold is expected to make more and more use of it. May as well be ready.
About IIS Applications

An IIS application is any file that is executed within a defined set of directories in a web site. Creating an application designates a particular directory in the web site as the starting point directory (also referred to as the application root). All files and directories under that starting point directory are considered part of that application until another starting point directory is encountered. See the About Applications topic in the IIS documentation.

Different components from the same .map file can be used with different map server pages. However, all map server pages must be part of the same Application.

About Object Pooling

Manifold components support object pooling, which avoids the overhead of creating new objects in highly loaded servers. Normally, when a process needs an object it is created on the fly using the component's class factory to create a new object. When object pooling is turned on objects that are created are retained in a pre-built pool of objects. When an object is needed an object from the pool is activated if one is available. Pooled objects are returned to the pool when no longer needed.

Suppose 20 users at once hit the web site. The first time the server experiences this peak load enough objects to service the 20 users will be created. As load drops, objects will be returned to the object pool. Thereafter as long as the load stays under 20 simultaneous users no objects need to be created: they will simply be activated from the pool. If load grows again and peaks at 25 users, a few more additional objects will be created since the objects in the pool will not be enough to service demand. Thereafter, enough objects will be in the pool to serve up to 25 simultaneous requests.

Logging

Manifold IMS supports logging. To enable logging, edit the config.txt file for a page and add the following line:

    logfile = c:\mapserver.log

where "c:\mapserver.log" is the fully qualified path name to the desired log file. The log file will be written in CSV format using field names that are self-describing. ticks is the number of 1/1000ths of a second spent serving the image. It's safe to use the same log file for different instances of the map server.

Be careful when enabling logging. It is always easy to turn on a logging function and then forget it has been enabled until the log file grows to a huge size.

Security Tips

Microsoft IIS is a secure and highly reliable web server when correctly configured by the system administrator. Nonetheless, the many options made available within IIS to support the needs of experts can result in security problems if administered improperly. Because an IIS web server connected to Internet can be attacked by any Internet user from anywhere in the world, a few general IIS security measures should be applied to any web server connected to the Internet. These measures are not specific to Manifold IMS: they should be applied to every web server.

We recommend the following elementary steps:

- Use Windows 2003, 2000 or XP. Windows XP configures IIS by default with more security than does Windows 2000. Windows Server 2003 has the most default security.
- Use NTFS file systems.
- Do not allow anyone to work interactively, to read mail or to browse web sites on the machine that hosts your web site. If you must use the same machine interactively, make sure that user logins you employ do not have Administrator privileges. If a user with Administrator privileges just once opens a malevolent email attachment the web server itself could be infected with a wide variety of "backdoor" viruses or Trojan Horse attacks. Never read email or browse the web when logged in as Administrator.
- Install the latest Service Pack for Windows. Service packs include many security patches.
- Configure Windows Update to automatically download and install the latest Windows updates.
• **Very Important:** If you create a default_admin.asp page for your IMS site, **make sure** to change access permissions on it so that ordinary Internet browsers cannot use it. To do this, in Windows Explorer right click on the default_admin.asp file, choose Properties and choose the Security tab. Remove all entries for allowed user names except Administrator or whatever other login name you wish to allow access to the administration page. Apply the settings and press OK.

• Use an **Administrator** password that is called something other than "Administrator". Brute-force password cracking attacks have half the problem solved if they can assume there is a login called "Administrator" to attack.

• Use a seriously long (over 16 digits), randomly constructed password to make it difficult for brute-force password cracking assaults to guess your password.

• Install any supplemental security patches published by Microsoft after the last Windows Service Pack. Microsoft now has tools that can scan your system to determine if there are security patches published that have not yet been installed. Visit Microsoft TechNet at [http://www.microsoft.com/technet/default.asp](http://www.microsoft.com/technet/default.asp) and drill down to their Security pages for information on the latest security tools and patches.

• Install anti-virus software. Keep your anti-virus software updated with regular downloads from your vendor.

• Install Microsoft URLScan (a free download from Microsoft TechNet) or similar tool. URLScan is an ISAPI filtering tool for IIS that rejects a wide variety of malformed URL requests before they can hit IIS. It was published by Microsoft in 2001 and is an absolutely essential tool to prevent many different types of Denial of Service attacks. No doubt the functions provided by URLScan will continue to be offered by Microsoft either as independent tools or built into IIS. Manifold IMS works perfectly with URLScan.

• Apply Microsoft's IIS Lockdown Tool (a free download from Microsoft TechNet), or its successors, using Advanced Lockdown. Uncheck the box that disables support for Active Server Pages (.asp) and otherwise accept all of the defaults suggested by Advanced Lockdown. Manifold IMS works perfectly with Advanced Lockdown settings so long as Active Server Pages are still enabled. This tool is also very easy to use. Some recent editions of IIS may be shipped by Microsoft in lockdown mode by default.

• Do not run unnecessary services. If your server runs a web site only, do not install or enable additional services such as FTP, NNTP or others. If such services are already installed, use Internet Services Manager to stop them. Do not install any accessory software or Windows components that are not required.

• Do not install unnecessary applications on your web server. Every application you install increases the **attack surface** that an adversary can exploit. For example, don't install things like Microsoft Office on a machine that will only be a web server.

• Do not use your web server, ever, for Internet browsing or other applications. Some people press into use as a web server an otherwise underutilized machine that is occasionally used (out of boredom while waiting for some administrative task to complete, etc.) to browse the web or for other purposes. Dedicate a machine to web serving and don't use it for other purposes.

• Do not connect your web server on a local area network to other machines unless you really know what you are doing with security. It is safest to keep it isolated and to use a USB drive to exchange files with other machines.

• If you use Remote Desktop to work with your server, change the port it uses to some port other than the default. (Don't forget to leave this port open in your firewall.)

• Learn to use Advanced TCP/IP settings for the Internet Protocol in Network properties for the network card in use. If your web site is intended only for specific machines (such as those on an Intranet), use IP security to deny access to any unauthorized machines (IP addresses). If you are connected to Internet, use TCP/IP filtering to permit only those TCP ports absolutely required to service your web site (typically, port 80 for HTTP and port 443 for HTTPS). Strict IP filtering will not allow services such as DNS (used if you are browsing from the web server) so a truly secure regime will limit utilization of the system to a web serving only.

• Install a firewall such as Windows Firewall and learn to use it. Use a hardware firewall and/or software solution such as Microsoft's Internet Security and Acceleration Server. Hardware firewalls now cost well under $100. Configure your hardware firewall so that only those ports required for serving web pages are enabled, with all other ports disabled.

• Install the latest updates for Manifold System from [manifold.net](http://www.manifold.net).

• Get a good book on Windows and IIS security. Read it carefully and apply the recommendations. Surf the web and read the many free resources on Windows and IIS security.

• Participate in Manifold-L (email discussion list format) and the Georeference [Forum at http://forum.manifold.net](http://forum.manifold.net) for Manifold System users. Ask questions of your more experienced colleagues and as you gain expertise yourself give back to the community by assisting newbies with security questions.

**Very important:** Although Manifold IMS can work with Microsoft security tools for IIS such as URLScan and the IIS Lockdown Tool, it is possible that other IIS applications installed on your system cannot work with these tools or require modifications to the default settings of these tools. It is critically important to read the documentation for these security tools.
Minimizing Per-User Temp Files

When using a High isolation level in IIS, a new process will be created for each user accessing IMS. When this is done the multiple instances of IMS can not share data with each other and will be forced to create a separate set of temporary files for each user. This provides high isolation but results in a large number of temporary files in the system TEMP folder.

One way of reducing the number of temporary files created is to lower the isolation level applied to the web application that uses IMS so that multiple instances of IMS can operate in the same process. This does create a risk in that a crash of one instance of IMS will likely crash all other instances, but given that it is relatively difficult to crash IMS in the first place, this might be a good tradeoff in the case of well-debugged .asp files.

Troubleshooting

See the troubleshooting topic Problems with the Internet Map Server for detailed checklists of what might be wrong.

Note

You must be running Manifold System Professional Edition or greater to have the capabilities discussed in this topic. All Manifold editions except Personal Edition include the map server. If you are running Manifold System Personal Edition you will not have the capabilities discussed in this topic.

See Also

Map Server Overview
Creating a Web Site
IMS Queries
Publishing Multiple Pages
Problems with the Internet Map Server
Customization

Manifold may be customized to present user-specified choices in various dialogs, such as palettes and the projection dialog. Custom options are made available by adding .xml files to the Config folder within the main Manifold installation directory (located at C:\Program Files\Manifold System\Config by default).

In Windows 2000, Windows XP and later Windows operating systems, ordinary user accounts do not have permissions to modify files within the Program Files folder. If we are working in such systems we can still use the customization features of Manifold by altering the Config file location in Tools - Options - File Locations to a folder where we have write permissions.

Customization options allow customization of Manifold installations in many areas, to name a few of the most common examples:

- Custom point styles taken from True Type fonts installed on the system.
- Custom scalable styles for points, lines areas and labels.
- Custom palettes and themes are used to color drawings or surfaces.
- Custom units of measure
- Custom ellipsoids
- Custom datums

Custom datum grids for NTv2
- Custom coordinate system (projection) presets
- Custom layout templates for use with print layouts.
- Command filters are XML files that specify which commands in the Manifold menu structure are enabled or disabled. Add-Ins that add commands to Manifold through scripts and configure toolbars.
- Scripts may invoke and control user interface elements such as dialogs. See the User Interface Scripting topic.
- Customized Geocoding Data Extensions providing additional geocoding data sources.
- Customized Manifold Geocoding Servers using the Geocoding Server Interface to connect to web-based or other geocoding data servers.
- Customized Manifold Image Servers using the Image Server Interface to connect to web-based or other image sources.
- Manifold may be programmed in a variety of ways using scripting languages, SQL, web programming or custom applications written in popular programming languages: See the Programming Manifold topic.

Many customization features use XML to specify additions. See the topics above for detailed on XML syntax and rules. In general, the following rules apply:

- XML (extended markup language) is used to specify both original factory settings and any options.
- All .xml files located in the Config directory and all subdirectories thereof will be read by Manifold upon startup. Any valid specifications found in such files will be used.
- Manifold's own "factory" settings are stored within the system itself. The factory settings may be overridden by providing .xml files that use the same names as the factory entities, if the Override system data with custom config files option is turned on (it is off by default) in the Miscellaneous page in Tools - Options.
- DO NOT CHANGE the factory settings unless you are truly expert in Manifold and are ready to work without technical support from manifold.net.
- When creating custom .xml specifications, the name of each custom item must be unique. Manifold keeps track of each customizable item by its name.
- After adding any new XML specifications, restart Manifold so they are loaded and available in the system.
- It makes sense to organize any custom additions into separate .xml files, one for palettes that are added, another for units of measure and so on. However, Manifold will parse all the .xml files found in the Config directory for valid attributes of any type in each of the files. One could therefore create a single .xml file, perhaps called MyCustomizations.xml, in which all customizations such as additional palettes, units of measure, custom ellipsoids, custom datums and a few custom projection presets are all kept.
Manifold® System Release 8.00 User Manual

- If you have added any custom specifications in your .map files, you are responsible to make sure that the .xml files providing those custom specifications are installed on every machine on which those custom specifications will be used.
- If you have purchased a technical support product from Manifold, support for customization requires developer level support.
- A simple check against elementary mistakes is to double-click open an .xml file to view it in Internet Explorer. If it is reasonably correct, IE will be able to open it and to show the contents in text form.

When a custom specification (such as a palette) is used within a particular component, the information for that custom specification will be saved with the component so the component will display properly. However, if the required .xml is not present on the system on which the .map file is used, the specification will not be available for any further usage.

**Editing .xml Files**

.xml files may be edited with Notepad. They are simply ASCII text files containing XML tags (called attributes) similar to those found in ordinary HTML pages used on the web. For example, a file called MyAltitudes.xml might contain:

```xml
<xml>
<palette>
    <name>My Altitude Palette</name>
    <colors>
        <color>#ebeaba</color>
        <color>#e4e5a0</color>
        <color>#faebbe</color>
        <color>#fee2a8</color>
        <color>#ffc882</color>
    </colors>
</palette>
</xml>
```

The above specifies a simple palette of five colors. In addition to Notepad, some users may prefer to use an XML editor such as Microsoft's XML Notepad editor.

**Predefined Items**

Manifold includes "intrinsic" definitions for quantities that are used to define others. These root definitions are compiled into the product and do not appear in any .xml files. They are written below as they would appear if they were defined in XML:

```xml
<unit>
    <name>meter</name>
    <nick>m</nick>
    <nickArea>sq m</nickArea>
    <scale>1</scale>
</unit>

<unit>
    <name>degree</name>
    <nick>deg</nick>
    <nickArea>sq deg</nickArea>
    <scale>1</scale>
    <latLon>True</latLon>
</unit>

<ellipsoid>
    <name>WGS 84</name>
```
Customization

Also included are presets for "plain" coordinate systems such as for Latitude / Longitude, Orthographic and others with default parameters.

Hexadecimal Numbers

Numbers written in hexadecimal form are written in a base 16 number system. Some XML customization attributes in Manifold, such as those used for point styles taken from a True Type font, can use hexadecimal numbers as an alternative to decimal numbers. Some attributes, such as the <color> attribute in palettes require the use of hexadecimal numbers.

Hexadecimal numbering uses the digits 1 through F to represent the numbers 1 through 15. One counts from 0 to 255 in hexadecimal using the sequence 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, 1C, 1D, 1E, 1F, 20, 21, ..., E9, EA, EB, EC, ED, EE, EF, F1, F2, F3, F4, F5, F6, F7, F8, F9, FA, FB, FC, FD, FE, FF.

Hexadecimal numbers are convenient for use with numbers in computers because each digit can represent a full byte. Thus, two-byte quantities such as the integer numbers from 0 to 255 can be represented with two-digit hexadecimal numbers from 00 to FF. Thus when one sees a number such as 04A3FFCE one knows exactly the values of each byte that composes this eight-byte, 64-bit number. Hexadecimal numbers are widely used in the specification of RGB color values.

Since a hexadecimal number like "22" can appear to be a decimal numbers, hexadecimal numbers are normally prefixed with a "#" symbol when used in Manifold .xml specifications.

Tech Tip

Manifold cannot use any customizations if the .xml files do not contain XML that is exactly correct. A useful test before launching Manifold is to open any newly created or modified .xml file in Internet Explorer. Internet Explorer will show a correct .xml file in a simple text format. If Internet Explorer cannot parse the .xml file, Manifold won't be able to either.
Custom Point Styles

Manifold can use point styles from four different sources:

- Point styles that are built into Manifold.
- Point styles that are created from True Type fonts installed in Windows.
- Point styles created from image files.
- Point styles created from custom scalable styles.

This topic discusses the first three methods of customizing point styles. For use of custom scalable styles see the Custom Scalable Styles for Points, Lines, Areas and Labels topic.

Point styles that are created from True Type fonts or images are loaded into Manifold based upon directions specified within .xml files found in the Config folder. When Manifold launches it reads any .xml files that are placed in the Config folder and then opens True Type fonts or image files they mention to create new point styles. If an .xml file mentions a True Type font or image file that is not available in the Windows system, the point style thus specified will not be available and will be displayed as the default circle point style.

For example, an .xml file adding all characters in the particular font could be provided to customize the Manifold installation. This .xml file tells Manifold to create point styles based on the characters in the given font. If for some reason the specified font is not installed in Windows, the point styles created from this .xml file will not be available.

Point Styles Created from Fonts

Point styles created from True Type fonts may be used just like any point style, except that they are always created using the foreground color only. Users can add their own point styles from True Type fonts by installing a desired True Type font and then creating an .xml file in the following pattern:

<xml>
<style>
  <name>Name for this style</name>
  <font>Windows font name</font>
  <symbol>64</symbol>
  <type>point</type>
</style>
</xml>

Rules

- There can be any number of <style> ... </style> entries within a single XML file between the <xml> tag at the beginning of the file and the </xml> tag at the end of the file.
- The name string supplied in the <name> attribute is mandatory and must be unique. Many users will construct a unique name by appending the symbol number to the name of the font.
- The font name string supplied in the <font> attribute is mandatory and must be one of the font names installed in Windows.
- The number in the <symbol> attribute is mandatory and must be a number that corresponds to a valid character, either in decimal notation or in hexadecimal (hexadecimal numbers must be preceded by a # character).
- The value in the <type> attribute is mandatory and must be "point".

Place the new or modified .xml file containing the point styles into the Config folder for Manifold (normally C:\Program Files\Manifold System Professional\Config) and restart Manifold. When Manifold launches, the system will scan all .xml files in the Config folder. Any <style> ... </style> entries of <type> "point" found in any of those .xml files that correspond to characters in installed Windows fonts will be loaded into the system as an available point style.

Example
We’ve installed the Microsoft Webdings font in our Windows system. We then created a file in the Config folder called PointsWebdings.xml that contains:

```xml
<xml>
  <style>
    <name>Webdings #2A</name>
    <font>Webdings</font>
    <symbol>#2A</symbol>
    <type>point</type>
  </style>
  <style>
    <name>Webdings #2B</name>
    <font>Webdings</font>
    <symbol>#2B</symbol>
    <type>point</type>
  </style>
</xml>
```

This file uses hexadecimal notation for the character numbers, the same as used in the Windows Character Map utility program. It installs two point styles from the Webdings font.

When Manifold launches it will see the PointsWebdings.xml file in the Config folder and will read it to see what it contains. It will find two point styles in the file and will load them for use. The two styles used may be seen below:

The Webdings font is available for free download from Microsoft’s typography page at


This high quality font includes numerous symbols that will be useful as point styles in GIS applications.

**Decimal notation**

If desired we can write the above example using decimal numbers for the <symbol> values as:

```xml
<xml>
  <style>
    <name>Webdings 042</name>
    <font>Webdings</font>
    <symbol>42</symbol>
    <type>point</type>
  </style>
  <style>
    <name>Webdings 043</name>
    <font>Webdings</font>
    <symbol>43</symbol>
    <type>point</type>
  </style>
</xml>
```
This example is equivalent to the hexadecimal form since decimal 42 is the same as hexadecimal 2A and decimal 43 is the same as hexadecimal 2B. Note that the <name> specified for each style is the font followed by the character number in decimal or hexadecimal. We can name the style however we choose; however, most users soon tire of creating inventive names for different point styles and eventually will adopt a standard naming scheme like the one used above.

Using individual characters or entire fonts

Manifold’s .xml specification for point styles allows individual characters to be used from installed Windows fonts. It is often the case that only a few symbols in a font are useful to us in GIS, so the ability to specify only selected characters prevents the formatting menus from being overwhelmed by many styles that are not useful.

At times we might wish to simply use an entire font. It would be tedious to specify each of 255 styles by hand. A much faster method is to use the PointsWingdings.xml file provided as an example on the Manifold downloads site. Suppose we want to use all of the characters in a font called NatoSymbols that has been installed as a Windows font in our system.

We would proceed as follows:

To exploit PointsWingdings.xml as a template:

1. Copy PointsWingdings.xml and paste it as PointsNatoSymbols.xml
2. Open PointsNatoSymbols.xml with Notepad [Right click on the file and choose Open With… Notepad].
3. Replace "Wingdings" with "NatoSymbols" throughout the file.
4. Save the file

Since the original file listed all of the characters using a standard naming scheme, doing a search and replace using the new font name will create a new file for the NatoSymbols font.

Using a previously created .xml file as a template to create a new file for a different font is very fast. Even if we wish to only use a few characters out of a font it is usually faster to first use a template to create an .xml file that includes all of the characters as styles and to then delete those styles that are not wanted. Since few fonts with interesting symbols have all available slots full of characters it usually will be necessary to delete some of the point styles created by simply using a “full” font template like PointsWingdings.xml.

A reminder: point styles cannot be created from a font unless it is installed in the system. Use the Fonts selection in the Windows Control Panel to see fonts that are installed in your system.

Point Styles Created from Images

Point styles created from images may be used just like any point style, except that they are not resizable (and so may result in unexpected scaling effects when size on the monitor or on printed pages is compared) and do not use either foreground or background formatting colors. In addition, point styles created from images will not be shown in selection color when selected.

Users can add their own point styles from images by installing a desired image file (one image per file) and then creating an .xml file in the following pattern:

<xml>
<style>
  <name>Name for this style</name>
  <file>images\filename.png</file>
  <type>point</type>
</style>
</xml>
Customization

**Rules**

- There can be any number of `<style>` entries within a single XML file between the `<xml>` tag at the beginning of the file and the `</xml>` tag at the end of the file.
- The name string supplied in the `<name>` attribute is mandatory and must be unique.
- The path string in the `<file>` attribute is mandatory. It specifies the path to the image file from the configuration directory. In the pattern shown above a `filename.png` image file must be within an `images` directory that is in the `Config` directory.
- Image files may be in any of the following formats: BMP, GIF, JPG, PNG, TIFF, EXIF, WMF and EMF. Each image file contains one image, corresponding to one point style.
- The value in the `<type>` attribute is mandatory and must be "point".

Place the new or modified `.xml` file containing the image point style specifications into the `Config` folder for Manifold (normally `C:\Program Files\Manifold System Professional\Config`) and restart Manifold. When Manifold launches, the system will scan all `.xml` files in the `Config` folder. Any `<style>` entries of `<type>"point"` found in any of those `.xml` files that correspond to image files will be loaded as a new point style.

Images can contain pixels in transparent color. `.png` format is a good format to use to save images that contain invisible pixels (transparent color). `.png` is fast and efficient and a good choice overall for iconic images.

Point styles created from images will be centered in the center of the image. Point styles created from images are not resizable, since they use a fixed pattern of pixels.

**Example**

Suppose we create an `images` subdirectory in the `Config` folder. We place several image files in `.png` format in that folder. We create a `PointImages.xml` file in the `Config` folder that contains:

```xml
<xml>
  <style>
    <name>Manifold Icon</name>
    <file>images\mfd_icon.png</file>
    <type>point</type>
  </style>
  <style>
    <name>Blue Pin</name>
    <file>images\blue_pin.png</file>
    <type>point</type>
  </style>
  <style>
    <name>Purple Pin</name>
    <file>images\purple_pin.png</file>
    <type>point</type>
  </style>
  <style>
    <name>Red Pin</name>
    <file>images\Red_pin.png</file>
    <type>point</type>
  </style>
  <style>
    <name>Orange Pin</name>
    <file>images\orange_pin.png</file>
    <type>point</type>
  </style>
</xml>
```
The above XML adds point styles consisting of a Manifold logo style and a collection of pushpin images in various colors. The *mfd_icon.png* image looks like:

![mfd_icon.png](images\mfd_icon.png)

The *orange_pin.png* image looks like:

![orange_pin.png](images\orange_pin.png)

The illustrations above were created by importing each *.png* file as an image and then opening the image in an image window with the background layer turned off in the layers pane and the border layer turned on. The border shows the extent of the image, which would otherwise be difficult to see since both images use invisible pixels. Note that the *orange_pin.png* image consists mostly of invisible pixels. The other color pin images were created from the orange image by using hue / saturation to change the hue and then exporting the altered image to a *.png* file.

The two point styles above can be used to show how point styles using images are centered at the point location. To illustrate center positioning we've created a map of Europe that has a drawing layer with major cities as points. We have two copies of the cities layer in our map. One layer is formatted to use a small circle for each point. The other layer uses either the Manifold logo point style or the orange pin style.
When we use the Manifold logo point style we can see from the positions of the small white circle (from the other
cities layer) that the logo is centered on each city point. We can also see that it is a challenge to design iconic
images that work equally well when superimposed on any color background. The Manifold logo, for example,
uses a splash of yellow color that disappears in lightly colored backgrounds.

When we use the orange pin point style we see that the image is also centered on each point; however, because
the image consists mostly of invisible pixels the center of the image falls visually at the point of the "pin". This
provides a nice effect as if the pin is pushed into the map at the point we wish to mark.

Tech Tips

Manifold cannot use any customizations if the .xml files do not contain XML that is exactly correct. A useful test
before launching Manifold is to open any newly created or modified .xml file in Internet Explorer. Internet
Explorer will show a correct .xml file in a simple text format. If Internet Explorer cannot parse the .xml file,
Manifold won't be able to either.

To review fonts use the Character Map utility that is available in most Windows systems. From the Start button
choose Programs - Accessories - System Tools - Character Map. (Also available from a Windows Command
Prompt (DOS) window as charmap.exe). The Character Map utility will show the name of each font installed in
your system. Clicking on any character will display it and will show the character code (number) in hexadecimal in
status bar in the lower left corner of the Character Map window. A Character Map code of 0x6C is the same as
#6C, the "0x" prefix being another way, like the "#" prefix, of indicating the number is given in hexadecimal.

XML files read by Manifold from the Config directory can contain specifications for any mixture of customizable
entities. For example, a single file can contain point styles, palettes, a new unit of measure and so on. It is
usually easier to keep track of things if customizations are organized into separate files by what is being
customized.

Choose image-based point styles with care if maps will be printed, since the literal printing of most point style
images at the pixel resolutions typical of many printers will result in very small, almost invisible point symbols. To
print more visible point styles, import the image into Manifold and use resize to increase the size of the image,
export the image and then use that image in the point style. This may result in very large point styles on screen
but will allow printing as desired.

The examples above place images in a subdirectory called images within the Config directory. The images can
be wherever desired within the Config directory; however, keeping them neatly organized within subdirectories is
a good idea if many images will be used.
The "embossed" look of the countries in Europe was created by first making boundary lines for each area and then copying the boundary lines into a second layer. The boundary lines in the lower layer were formatted with a size of 2 and given a foreground color lighter than the areas. The boundary lines in the upper layer were formatted with a size of 1 and given a foreground color darker than the areas. This layer was then repositioned slightly with CTRL-grabber to the Northwest.

A "raised border" effect can be created by reversing the sizes and colors of the boundary lines in the two layers. If the upper (shifted) layer is the lighter, thicker layer and the lower line layer is the thinner, darker layer the perceptual effect is one of raised borders as seen above.

New releases of Manifold may introduce new formatting styles. The /slist command line option provides a handy way for programmers to get a comprehensive list of all area, label, line and point styles available and their names.
Custom Scalable Styles for Points, Lines, Areas and Labels

Manifold allows definition of scalable custom styles for points, lines, areas and labels. Scalable styles are defined using XML files that contain one or more style definitions. Style definitions are built up from a repertoire of vector painting commands.

A custom style is defined using a `<style>` tag. A scalable custom style must also have a `<code>` tag with an optional `autosize` attribute and one or more painting commands.

Scalable styles are defined by .xml files in the following pattern:

```xml
<xml>
  <style>
    <name>MyName</name>
    <type>point</type>
    <preview>8</preview>
    <code>
      (painting commands go here)
    </code>
  </style>
</xml>
```

A custom line style is created by defining the metric of a line segment with an X extent of -1000 to 1000, which is then repeated along the line.

Rules

There can be any number of `<style>` ... `<style>` entries within a single XML file between the `<xml>` tag at the beginning of the file and the </xml> tag at the end of the file. Within `<style>` entries the following attributes are used:

- `<name>` is mandatory and must be unique.
- `<type>` is mandatory and must be `area`, `label`, `line` or `point`, specifying what type of style this is.
- `<preview>` is optional and specifies the preview size of the style, in points.
- `<code>` is mandatory. It contains one or more painting commands that describe how the style is drawn ("painted"). It may contain an optional `autosize` attribute and one or more painting commands.
- There can one or more painting commands entries within a `<code>` entry.

Painting Commands

Painting commands that may be used within the `<code>` tag are:

- `<arc xmin=... ymin=... xmax=... ymax=... beg=... end=... fore=... />` - The `<arc>` tag paints an arc between the given "beg" and "end" angles. The angles are measured in degrees and can be fractional.
- `<dash size=... backP=... foreD=... backD=... fore=... back=.../>` - The `<dash>` tag paints a dashed line of given size. The size attribute defines the relative size of the object painted by the tag with respect to the size of the line object (1000 is the same size, 500 is half the size of the line object, etc). The backP attribute defines the length of the first dash, painted with background color, in 1/1000th of the size of the line object. The foreD attribute defines the length of the second dash, painted with foreground color. The backD attribute defines the length of the third dash, painted with background color. The summary length of the dashes painted with each color must be positive. The length of each dash must be non-negative.
- `<ellipse xmin=... ymin=... xmax=... ymax=... fore=... back=... />` - The `<ellipse>` tag paints an ellipse.
- `<polygon fore=... back=...> <point x=... y=... /> ... </polygon>` - The `<polygon>` tag paints a polygon. Each `<polygon>` tag should contain three or more `<point>` tags.
- The `<polyline>` tag paints a polyline. Each `<polyline>` tag should contain two or more `<point>` tags.

- The `<rectangle>` tag paints a rectangle.

- The `<refLine>` tag paints a line object of given size and style. The size attribute defines the relative size of the object painted by the tag with respect to the size of the line object (1000 is the same size, 500 is half the size of the line object, etc). The style attribute must contain the name of an already defined line style.

- The `<refPoint>` tag paints a point object of given size and style. The style attribute of the `<refPoint>` tag must contain the name of an already defined point style. The `angle` attribute gives the rotation angle of the point symbol in degrees (clockwise rotation).

- The `<text>` tag paints the text of the context label object with given style. The `style` attribute of the `<text>` tag can be any of the following values, which correspond to Manifold label styles: `box`, `boxB`, `boxCorners`, `boxFrame`, `boxGradHorzDD`, `boxGradHorzDL`, `boxGradHorzLL`, `boxGradVertDD`, `boxGradVertDL`, `boxGradVertLL`, `boxInner`, `boxL`, `boxR`, `boxT`, `boxThick`, `boxThickB`, `boxThickL`, `boxThickR`, `boxThickT`, `halo`, `haloLB`, `haloLT`, `haloRB`, `haloRT`, `haloWide`, `plain`.

All attributes are required except the `back` and `fore` attributes for all tags and the `border` and `style` attributes in the `<text>` tag. The possible values of the `back`, `border` and `fore` attributes are `b` (background color), `f` (foreground color), `m` (a mix between the background color and the foreground color), `n` (transparent color), or an absolute color value (such as `#00ff00` or `0:255:0`).

See the Custom Palettes and Themes topic for notations that may be used to specify absolute color value in either hexadecimal or decimal form.

By default, a style is defined for an object with a size of 1 (1 printer's point). If the `autosize` attribute in the `<code>` tag is set to `true` (the default), the system scales the output of painting commands so that a box of (X: -1000...1000, Y: -1000...1000) corresponds to the box used by the built-in symbol of the same size. If the `autosize` attribute is set to `false`, the system scales the output of the painting commands so that 1000 X or Y units equal X physical points, where X is the size of the object being painted.

**Examples**

**A custom point style:**

```xml
<xml>
  <style>
    <name>customFlag</name>
    <type>point</type>
    <code>
      <polygon>
        <point x="0" y="500" />
        <point x="1000" y="750" />
        <point x="0" y="1000" />
      </polygon>
    </code>
  </style>

  <style>
    <name>customFlagWithDot</name>
    <type>point</type>
    <code>
      <ellipse xmin="-200" ymin="-200" xmax="200" ymax="200" back="m" />
      <refPoint x="0" y="0" size="1000" style="customFlag" />
    </code>
  </style>
</xml>
```
A custom line style:

```xml
<xml>
<style>
  <name>Custom1</name>
  <type>line</type>
  <preview>8</preview>
  <code>
    <polyline>
      <point x="0" y="0"/>
      <point x="0" y="400"/>
    </polyline>
    <polyline>
      <point x="0" y="0"/>
      <point x="200" y="0"/>
      <point x="200" y="200"/>
    </polyline>
    <polyline>
      <point x="200" y="0"/>
      <point x="400" y="0"/>
      <point x="400" y="200"/>
    </polyline>
    <polyline>
      <point x="400" y="0"/>
      <point x="600" y="0"/>
      <point x="600" y="200"/>
    </polyline>
    <polyline>
      <point x="600" y="0"/>
      <point x="800" y="0"/>
      <point x="800" y="200"/>
    </polyline>
    <polyline>
      <point x="800" y="0"/>
      <point x="1000" y="0"/>
      <point x="1000" y="200"/>
    </polyline>
  </code>
</style>
<style>
  <name>Custom2</name>
  <type>line</type>
  <preview>8</preview>
  <code>
    <refLine style="Custom1" size="1000"/>
    <ellipse xmin="-100" ymin="-300" xmax="100" ymax="-100"/>
  </code>
</style>
</xml>
```
A custom area style:

```xml
<xml>
  <style>
    <name>Weird Hatch</name>
    <type>area</type>
    <code>
      <polyline>
        <point x="-1000" y="0" />
        <point x="800" y="600" />
        <point x="-500" y="900" />
        <point x="1000" y="0" />
      </polyline>
    </code>
  </style>
</xml>
```

**Tech Tip**

Each built-in style for areas, labels, lines and points has been assigned a name, for use in custom styles. For a complete list of names, see the manifold.net web site within the Support pages hierarchy.

**See Also**

Customization

Custom Point Styles
Custom Palettes and Themes

Manifold has two systems to allow specification of custom formatting characteristics such as color in thematic formatting. One system, using custom palettes, is a legacy system that can be used to specify only colors. The more general method is to use custom themes, which can specify many different formatting characteristics such as rotation angles, color, and the size and style of labels, areas, lines or points.

Because custom palettes may not be supported at some time in the future, users should take care to employ only custom themes. The information in this topic on custom palettes is provided for backward compatibility with earlier versions of Manifold.

Note: in this topic, the word “theme” is used to mean a collection of formatting characteristics used with values for thematic formatting. It is a different use of the same word that is also used to mean a theme component.

In both cases, either custom palettes or themes used for thematic formatting of drawings and surfaces or terrains are loaded from .xml files found in the Config folder. Specifications for custom palettes or themes are enclosed within <palette> … </palette> tags for a custom palette or <theme> … </theme> tags to specify a custom theme.

If an .xml file specifying a custom theme occurs in the Config folder Manifold will automatically load it upon startup. If the .xml file specifying the custom theme neglects to provide a name for the theme, Manifold will name the theme Custom X, where X is a number that makes the name unique, for example, Custom 1, Custom 2 and so on.

Custom Palettes

Manifold palettes come in two forms: relative palettes in which colors only are specified, and fixed palettes that also specify values for the intervals to which colors apply. In relative palettes, only a list of colors need be specified. In fixed palettes, a list of colors as well as a corresponding list of values must be specified.

Users can add their own palettes (including those with fixed values) by creating .xml files in the following patterns:

For relative palettes:

```xml
<xml>
<palette>
  <name>My Relative Palette</name>
  <colors>
    <color>#000000</color>
    <color>#ff0000</color>
    <color>#00ff00</color>
    <color>#0000ff</color>
    <color>#ffffff</color>
  </colors>
</palette>
</xml>
```

For fixed palettes:

```xml
<xml>
<palette>
  <name>My Fixed Palette</name>
  <colors>
    <color>#000000</color>
    <color>#ff0000</color>
    <color>#00ff00</color>
    <color>#0000ff</color>
    <color>#ffffff</color>
  </colors>
</palette>
</xml>
```
Either fixed or relative palettes may also use decimal numbers to specify RGB color values as well as hex numbers:

```xml
<xml>
<palette>
  <name>Sample palette</name>
  <colors>
    <color>0</color>
    <color>255:0:0</color>
    <color>0:255:0</color>
    <color>0:0:255</color>
    <color>65280</color>
    <color>#000000</color>
  </colors>
</palette>
</xml>
```

The example above decimal numbers with the last color specified using hexadecimal. Decimal and hexadecimal color specifications may be used within the same palette.

### Rules for Palettes

- There can be any number of `<palette>` ... `<palette>` entries within a single XML file between the `<xml>` tag at the beginning of the file and the `</xml>` tag at the end of the file.
- The name string in the `<name>` attribute is mandatory and must be unique.
- Values of colors are specified in the `<color>` attribute in either hexadecimal form or decimal form.
- Hexadecimal colors are in "#rrggbb" form where rr, gg and bb are pairs of hexadecimal digits giving a hexadecimal number corresponding to the decimal number range from 0 to 255 for Red (rr), Green (gg) or Blue (bb). Leading zeros are significant, so that RGB values of 3, 5 and 6 are coded as 00003 in hexadecimal numbers. The examples above show a list of colors that are black, pure red, pure green, pure blue and white.
- Decimal colors may be in two forms: either r:g:b form where r, g and b are decimal integer numbers in the range from 0 to 255 or as a single decimal number that is the decimal equivalent of a #rrggbb hexadecimal number.
- There must be at least two colors listed in a palette.
- Fixed palettes must also include a list of values in the `<values>` attributes.
- Values in the `<value>` attributes are floating-point numbers. Negative numbers are prefixed with a - minus sign.
- Values must be listed in ascending order from lowest to highest value. Each value must be unique within the list for that palette. The number of values must be exactly the same as the number of colors in that palette.

### Custom Themes
Users can add their own themes by creating .xml files in the following pattern:

```xml
<xml>
  <theme>
    <type>color</type>
    <interval>true</interval>
    <colors>
      <color>#333333</color>
      <color>#666666</color>
      <color>#999999</color>
      <color>#cccccc</color>
      <color>#ffffff</color>
    </colors>
    <colorMin>#000000</colorMin>
    <colorMax>#ffffff</colorMax>
    <colorDef>#000000</colorDef>
    <values>
      <value>-69.00</value>
      <value>-21.00</value>
      <value>27.00</value>
      <value>75.00</value>
      <value>123.00</value>
    </values>
  </theme>
</xml>
```

Themes use a more modern nomenclature than do palettes. Instead of referring to a "relative" or "fixed" palette, themes are called interval based (equivalent to relative palettes) or value based (equivalent to fixed palettes).

**Rules**

- The `<theme>` tag surrounds the definition of the theme. There can be any number of `<theme>` entries within a single XML file between the `<xml>` tag at the beginning of the file and the `</xml>` tag at the end of the file. However, only the first such tag is used. The others will be ignored.
- The `<type>` tag contains the type of the theme and can be equal to any of the following values: angle, color, areaSize, areaStyle, labelSize, labelStyle, lineSize, lineStyle, pointSize, or pointStyle.
- The `<interval>` tag determines whether the theme is interval-based (true) or value-based (false). Interval-based themes assign formatting to data falling within the specific intervals, and value-based themes assign formatting to data equal to specific values.
- The `<values>` and `<value>` tags contain interval breaks or unique values depending on whether the theme is interval-based or value-based. The theme must contain at least one value.
- The `<column>` tag contains the name of the column to use. This tag is optional.
- Tags specifying specific characteristics consist of a tag using the plural name of the given type followed by singular name tags for each specific case. The `<colors>` and `<color>` tags contain formatting data for colors as seen in the prototype XML above. A theme for line styles would use `<lineStyles>` and `<lineStyle>` tags, a theme for point styles would use `<pointStyles>` and `<pointStyle>` tags and so on. The number of `<color>` tags (or other singular tags) and `<value>` tags must be the same.
- The `<colorMin>` and `<colorMax>` tags are used with interval-based themes to format color and contain formatting data for values that are less than the lowest break, or greater than the highest break. An interval-based theme used to format line styles would use `<lineStyleMin>` and `<lineStyleMax>` tags and so on.
- The `<colorDef>` tag is used with value-based themes and contains formatting data to use for values that are different from any of defined unique values. A theme for line styles would use the `<lineStyleDef>` tag and so on.
- Values of colors are specified in the `<color>` attribute in either hexadecimal form or decimal form.
- Hexadecimal colors are in "#rrggbb" form where rr, gg and bb are pairs of hexadecimal digits giving a hexadecimal number corresponding to the decimal number range from 0 to 255 for Red (rr), Green (gg) or Blue (bb). Leading zeros are significant, so that RGB values of 3, 5 and 6 are coded as 030506 in...
hexadecimal numbers. The examples above show a list of colors that are black, pure red, pure green, pure blue and white.

- Decimal colors may be in two forms: either \texttt{r:g:b} form where \(r\), \(g\) and \(b\) are decimal integer numbers in the range from 0 to 255 or as a single decimal number that is the decimal equivalent of a \#rrggbb hexadecimal number.
- There must be at least two colors listed in a theme.
- Fixed themes must also include a list of values in the \texttt{<values>} attributes.
- Values in the \texttt{<value>} attributes are floating-point numbers. Negative numbers are prefixed with a - minus sign.
- Values must be listed in ascending order from lowest to highest value. Each value must be unique within the list for that theme. The number of values must be exactly the same as the number of colors in that theme.

**Theme Examples**

To create an example of a theme, use the Thematic Formatting dialog to create a format and then press the Save to File button to save it as an .xml file. Open the .xml file with Notepad to see how the .xml implements the above rules.

**Using Palettes or Themes**

To utilize newly created or modified palettes or themes place the new or modified .xml file containing the palettes into the Config folder for Manifold (normally C:\Program Files\Manifold System\Config) and restart Manifold.

To quickly check what palettes are currently available, see the Tools - Options, Color Palettes dialog.

When Manifold launches, the system will scan all .xml files in the Config folder. Any \texttt{<palette>} or \texttt{<theme>} entries found in any of those .xml files will be loaded into the system as available palettes.

**Color Specification in Hex or Decimal**

Most classic graphics programming is done specifying colors using hex numbers in the \#rrggbb pattern. However, most Windows programs used in graphics editing report RGB colors using triplets of decimal numbers. Manifold can accept both numeric representation styles.

The hex specifications:

\begin{verbatim}
<color>#000000</color>
<color>#ff0000</color>
<color>#00ff00</color>
<color>#0000ff</color>
\end{verbatim}

...define exactly the same colors as the decimal specifications:

\begin{verbatim}
<color>0:0:0</color>
<color>255:0:0</color>
<color>0:255:0</color>
<color>0:0:255</color>
\end{verbatim}

The third style, of reporting concatenated hex values (\#rrggbb) as their equivalent decimal number value is rarely used except as a shorthand way of specifying black color. The following pairs of values are equivalent:

\begin{verbatim}
<color>#000000</color>
<color>0</color>
<color>#ff0000</color>
<color>6711680</color>
\end{verbatim}
Example

We've created a file in the Config folder called mypalettes.xml that contains:

```xml
<xml>
<palette>
  <name>Atlas Colors in Meters</name>
  <colors>
    <color>#7897bb</color>
    <color>#84afcf</color>
    <color>#95c4dc</color>
    <color>#bfdbe3</color>
    <color>#e7eff1</color>
    <color>#b6d294</color>
    <color>#d5e3b9</color>
    <color>#f2dbb4</color>
    <color>#e7bf83</color>
    <color>#deac6b</color>
    <color>#ab8e59</color>
    <color>#e2ddd6</color>
    <color>#ebe7e2</color>
    <color>#f2f0ee</color>
    <color>#ffffff</color>
  </colors>
  <values>
    <value>-8000</value>
    <value>-6000</value>
    <value>-4000</value>
    <value>-2000</value>
    <value>-200</value>
    <value>0</value>
    <value>100</value>
    <value>200</value>
    <value>500</value>
    <value>1000</value>
    <value>2000</value>
    <value>3000</value>
    <value>4000</value>
    <value>5000</value>
    <value>6000</value>
  </values>
</palette>
<palette>
  <name>Atlas Colors in Feet</name>
</xml>
```
<colors>
  <color>#7897bb</color>
  <color>#84afcf</color>
  <color>#95c4dc</color>
  <color>#b6d294</color>
  <color>#d5e3b9</color>
  <color>#f2dbb4</color>
  <color>#e7bf83</color>
  <color>#deac6b</color>
  <color>#ab8e59</color>
  <color>#e2ddd6</color>
  <color>#ebe7e2</color>
  <color>#ff00ee</color>
  <color>#ffffff</color>
</colors>

/values>
  <value>-26248</value>
  <value>-19686</value>
  <value>-13124</value>
  <value>-6562</value>
  <value>-656</value>
  <value>0</value>
  <value>328</value>
  <value>656</value>
  <value>1640</value>
  <value>3281</value>
  <value>6562</value>
  <value>9843</value>
  <value>13124</value>
  <value>16409</value>
  <value>19686</value>
</values>
</palette>

<palette>
  <name>Atlas Colors</name>
  <colors>
    <color>#7897bb</color>
    <color>#84afcf</color>
    <color>#95c4dc</color>
    <color>#b6d294</color>
    <color>#d5e3b9</color>
    <color>#f2dbb4</color>
    <color>#e7bf83</color>
    <color>#deac6b</color>
    <color>#ab8e59</color>
    <color>#e2ddd6</color>
    <color>#ebe7e2</color>
    <color>#ff00ee</color>
    <color>#ffffff</color>
  </colors>
</palette>
When Manifold launches it will see the `mypalettes.xml` file in the `Config` folder and will read it to see what it contains. It will find three palettes in the file and will load them for use. Two of the palettes are fixed palettes while the third file is a relative palette. Note that negative numbers may be used as values for fixed palettes.

All three palettes use the same colors. The two fixed palettes have different values, which we can see upon examination are the same numbers that would be used for elevations either in feet or in meters. This is a fairly typical arrangement when creating a new palette intended for use with surfaces. Surfaces usually originate in either meter-based data sets, where the elevation values are given in meters, or in foot-based elevation values such as US SDTS DEMs where the elevations are in feet. It is convenient to have the same fixed palette available with both meter-based as well as foot-based palettes so that the color scheme can be used for the same elevation above or below sea level regardless of whether the numbers in the surface represent meters or feet.

It is also convenient to have a version of a fixed palette available as a relative palette. This is allows a quick look at how the full range of palette colors appears in use. We can use whatever data set we want and even if it only ranges between -100 and 1000 we will still see the full set of colors.

**Tech Tip**

Manifold cannot use any customizations if the `.xml` files do not contain XML that is exactly correct. A useful test before launching Manifold is to open any newly created or modified `.xml` file in Internet Explorer. Internet Explorer will show a correct `.xml` file in a simple text format. If Internet Explorer cannot parse the `.xml` file, Manifold won’t be able to either.
Custom Units of Measure

Manifold uses units of measure in a variety of places such as in the units box used with most projections. The definitions of units available to Manifold are compiled into the product. User-supplied definitions may be loaded from .xml files found in the Config folder. If desired, users can add their own units of measure to incorporate specialized, historical or local units of measure into Manifold.

Units are defined by .xml files in the following pattern:

```xml
<xml>
  <unit>
    <name>My Own Centimeter</name>
    <nick>my cm</nick>
    <nickArea>my sqcm</nickArea>
    <scale>0.01</scale>
  </unit>
  <unit>
    <name>Five-degree</name>
    <nick>fd</nick>
    <nickArea>my sqfd</nickArea>
    <scale>5.0</scale>
    <latLon>True</latLon>
  </unit>
</xml>
```

**Rules**

- There can be any number of `<unit>` entries within a single XML file between the `<xml>` tag at the beginning of the file and the `</xml>` tag at the end of the file.
- The name string supplied in the `<name>` attribute is mandatory and must be unique.
- The flag in the `<latLon>` attribute must be set to "True" for angular measurement units such as degrees or radians and set to "False" or omitted for ordinary measurement units.
- The value in the `<scale>` attribute is the size of the unit relative to a degree (if the latLon flag is True) or a meter (if the latLon flag is False).
- `<nick>` and `<nickArea>` names are nicknames (abbreviations) of the units of measure that will be used by Manifold in various readouts. The `<nickArea>` name is used for the areal unit of measure, such as square meters.

To utilize newly created or modified units place the new or modified .xml file containing the units into the Config folder for Manifold (normally C:\Program Files\Manifold System Professional\Config) and restart Manifold.

When Manifold launches, the system will scan all .xml files in the Config folder. Any `<unit>` entries found in any of those .xml files will be loaded into the system as available units.

**Example**

We've created a file in the Config folder called myunits.xml that contains units we would like to use with Manifold. We find it annoying that English units like feet are derived from the physical dimensions of an English king's body parts, so we will create new units based on our own body parts. We also find working in units of five degrees more convenient than just one degree at a time so we will create a new angular unit that is equivalent to five degrees:

```xml
<xml>
  <unit>
    <name>Thumb</name>
    <nick>thb</nick>
    <nickArea>sq thb</nickArea>
  </unit>
</xml>
```
When Manifold launches it will see the myunits.xml file in the Config folder and will read it to see what it contains. It will find definitions for three new units of measure the file and will load them for use. Two of the units are linear measures defined from meters while the third is an angular measurement based on degrees.

One Thumb is equivalent to 0.05 meters, or five centimeters. One Arm is equivalent to 50 centimeters. Note that both of these units of measures are linear measures since the latLon attribute is not present in them.

The definition for ArcHand says it is equivalent to five degrees. We have decided to call our unit an "ArcHand" because a hand has five fingers just like this unit represents five degrees. When reported on the status bar in drawings projected using ArcHands the abbreviation will be ah or sq ah.

**Tech Tip**

Manifold cannot use any customizations if the .xml files do not contain XML that is exactly correct. A useful test before launching Manifold is to open any newly created or modified .xml file in Internet Explorer. Internet Explorer will show a correct .xml file in a simple text format. If Internet Explorer cannot parse the .xml file, Manifold won't be able to either.
Custom Ellipsoids

Manifold uses ellipsoids in projection calculations. See the The Earth as an Ellipsoid for more information. The definitions of ellipsoids available to Manifold are compiled into the product. User-supplied definitions may be loaded from .xml files found in the Config folder. If desired, users can add their own ellipsoids to incorporate specialized, historical or local ellipsoids into Manifold.

Ellipsoids are defined by .xml files in the following pattern:

<xml>
    <ellipsoid>
        <name>My Ellipsoid</name>
        <majorAxis>6377563</majorAxis>
        <invFlattening>299</invFlattening>
    </ellipsoid>
</xml>

Rules

- There can be any number of <ellipsoid> ... </ellipsoid> entries within a single XML file between the <xml> tag at the beginning of the file and the </xml> tag at the end of the file.
- The name string supplied in the <name> attribute is mandatory and must be unique.
- <majorAxis> is mandatory and is the length of the major axis in meters.
- <invFlattening> is the value of the inverse flattening. The <invFlattening> attribute may be replaced with by the <eccentricity> attribute specifying the eccentricity of the ellipsoid, or by the <minorAxis> attribute giving the length of the minor axis in meters. A value for the minor axis that is greater than the major axis will be rejected.

To utilize newly created or modified ellipsoids place the new or modified .xml file containing the ellipsoids into the Config folder for Manifold (normally \C:\Program Files\Manifold System Professional\Config) and restart Manifold.

When Manifold launches, the system will scan all .xml files in the Config folder. Any <ellipsoid> ... </ellipsoid> entries found in any of those .xml files will be loaded into the system as available ellipsoids.

Example

We've created a file in the Config folder called myellipsoids.xml that contains ellipsoids we would like to use with Manifold. These ellipsoids will help us create maps of the Moon and Mars:

<xml>
    <ellipsoid>
        <name>Moon</name>
        <majorAxis>1738000</majorAxis>
        <eccentricity>0.058730</eccentricity>
    </ellipsoid>
    <ellipsoid>
        <name>Mars</name>
        <majorAxis>3388000</majorAxis>
        <minorAxis>3388000</minorAxis>
    </ellipsoid>
</xml>

When Manifold launches it will see the myellipsoids.xml file in the Config folder and will read it to see what it contains. It will find definitions for ellipsoids named Moon and Mars. The Moon is treated as a true ellipsoid and is defined using a major axis and eccentricity. Mars is a sphere and defined by using the same values for the major axis and the minor axis. The ability to specify an ellipsoid using eccentricity or inverse flattening or minor
axis is very handy since the definitions of ellipsoids that are available to us will likely have one of these three parameters in addition to the major axis.

See any standard text on computational cartography for detailed discussions of eccentricity and inverse flattening. Inverse flattening is defined as \( \frac{\text{majorAxis}}{\text{major- minorAxis}} \). Eccentricity is defined as the square root of \( 1 - \left( \frac{\text{minorAxis}}{\text{majorAxis}} \right)^2 \).

**Tech Tip**

Manifold cannot use any customizations if the .xml files do not contain XML that is exactly correct. A useful test before launching Manifold is to open any newly created or modified .xml file in Internet Explorer. Internet Explorer will show a correct .xml file in a simple text format. If Internet Explorer cannot parse the .xml file, Manifold won't be able to either.
Custom Datums

Manifold uses datums in projection calculations. A datum is an ellipsoid that is combined with an offset of the ellipsoid relative to the Earth's center (defined as the center of the WGS84 datum). The definitions of datums available to Manifold are compiled into the product. User-supplied definitions may be loaded from .xml files found in the Config folder. If desired, users can add their own datums to incorporate other datums into Manifold.

Datums are defined by .xml files in the following pattern:

```xml
<xml>
  <datum>
    <name>My Datum</name>
    <ellipsoid>My Ellipsoid</ellipsoid>
    <centerX>0</centerX>
    <centerY>12</centerY>
    <centerZ>28</centerZ>
    <rotationX>0</rotationX>
    <rotationY>0</rotationY>
    <rotationZ>5</rotationZ>
    <scaleAdjustment>0</scaleAdjustment>
  </datum>
</xml>
```

Rules

- There can be any number of `<datum>` entries within a single XML file between the `<xml>` tag at the beginning of the file and the `</xml>` tag at the end of the file.
- The name string supplied in the `<name>` attribute is mandatory and must be unique. If the `<ellipsoid>` tag is used it must contain the name of an existing ellipsoid (user-defined ellipsoids are OK). The `<ellipsoid>` tag is not mandatory and can be replaced by an alternative specification of the ellipsoid using equivalent sets of tags, such as `<majorAxis>` and `<minorAxis>`, `<majorAxis>` and `<eccentricity>` or `<majorAxis>` and `<invFlattening>` tags.
- When using alternate specifications for ellipsoids, the `<majorAxis>` tag is mandatory and is the length of the major axis in meters. `<invFlattening>` is the value of the inverse flattening. The `<invFlattening>` attribute may be replaced with by the `<eccentricity>` attribute specifying the eccentricity of the ellipsoid, or by the `<minorAxis>` attribute giving the length of the minor axis in meters. A value for the minor axis that is greater than the major axis will be rejected.
- `<centerX>`, `<centerY>` and `<centerZ>` tags specify the offsets of the datum center along the X, Y and Z axis with respect to the center of WGS84, and are optional. The offset distances are in meters.
- `<rotationX>`, `<rotationY>` and `<rotationZ>` tags specify the rotation of the datum near the X, Y and Z axis with respect to WGS84, and are optional. The rotation angles are in whatever units are specified by the `unit` attribute. If no unit is specified, the rotation angles are in degrees.
- `<scaleAdjustment>` specifies the adjustment of the datum scale relative to WGS84, in units per million. That is, a scale adjustment of 1 means that the scale of the datum is 1.000001 that of WGS84.

To utilize newly created or modified datums place the new or modified .xml file containing the datums into the Config folder for Manifold (normally C:\Program Files\Manifold System Professional\Config) and restart Manifold.

When Manifold launches, the system will scan all .xml files in the Config folder. Any `<datum>` entries found in any of those .xml files will be loaded into the system as available datums.

Example

We've created a file in the Config folder called mydatums.xml that contains a new datum we would like to use with Manifold. The file contains:

```xml
<xml>
  <datum>
    <name>My Datum</name>
    <ellipsoid>My Ellipsoid</ellipsoid>
    <centerX>0</centerX>
    <centerY>12</centerY>
    <centerZ>28</centerZ>
    <rotationX>0</rotationX>
    <rotationY>0</rotationY>
    <rotationZ>5</rotationZ>
    <scaleAdjustment>0</scaleAdjustment>
  </datum>
</xml>
```
<name>My Datum</name>
<ellipsoid>Manifold 2001</ellipsoid>
<centerX>121</centerX>
<centerY>328</centerY>
<centerZ>15</centerZ>
</datum>
</xml>

When Manifold launches it will see the mydatums.xml file in the Config folder and will read it to see what it contains. It will find a definition for a datum called My Datum that uses an ellipsoid called Manifold 2001. If this ellipsoid is not part of the standard set of ellipsoids in Manifold, it must be defined elsewhere in an <ellipsoid> specification in one of our customizing .xml files.

Tech Tip

Manifold cannot use any customizations if the .xml files do not contain XML that is exactly correct. A useful test before launching Manifold is to open any newly created or modified .xml file in Internet Explorer. Internet Explorer will show a correct .xml file in a simple text format. If Internet Explorer cannot parse the .xml file, Manifold won't be able to either.

See Also

See the The Earth as an Ellipsoid for more information.
Custom Datum Grids for NTv2

Manifold allows definition of custom datum grids for the NTv2 transform. Datum grids are stored in GSB "grid shift" files and are described using XML files with the same name that contain one or more datum pairs to use with the grid.

The GSB files and XML files must be placed in the folder specified in the Tools - Options file location Datum Grids value. By default, this file location specifies a Grids folder within the main Manifold System installation folder.

XML File Format for Datum Transformation Pairs

A datum pair is defined using a <transform> tag. There can be more than one <transform> tag in an XML file for the GSB file.

The definition of a datum pair is done in the following pattern:

```xml
<xml>
  <transform>
    <source>Datum Name 2</source>
    <target>Datum Name 1</target>
  </transform>
</xml>
```

Both <source> and <target> tags specify the names of datums, and are mandatory.

The point of providing an XML file like the above is to tell Manifold the names of the datums that are to be converted with NTv2 using this grid shift file. Since NTv2 transformations are done only for a specific region, this assures that only those datums the names of which exactly match the datum names specified will be transformed using NTv2. Other datums will be transformed using the high accuracy, general purpose routines within Manifold.

Given a pair of source and target datums, Manifold will use the custom grid file to convert coordinates both from the source datum to the target datum and from the target datum to the source datum. Switching the order of the source and target datums will cause loss of precision.

Example

Suppose we want to use the NB7783V2.GSB grid shift file for converting from ATS77 to NAD83(CSRS) used in New Brunswick, Canada. We would provide an XML file named NB7783V2.xml that contained:

```xml
<xml>
  <transform>
    <source>ATS77</source>
    <target>NAD83(CSRS)</target>
  </transform>
</xml>
```

We would place both the NB7783V2.GSB file and the accompanying NB7783V2.xml file into a folder called Grids within the Manifold installation folder, or such other folder as we care to specify in the Tools - Options file location Datum Grids value.

When Manifold starts up, it will locate the .gsb file and accompanying .xml file and will load the ability to convert from ATS77 to NAD83(CSRS) using the grid file provided with NTv2.

See Also
Customization

Projections
Custom Coordinate System (Projection) Presets

Manifold includes a huge number of projections in the Projections pane. Each projection is really the definition of a coordinate system. Many coordinate systems are defined by standard formulas in which given parameters may be varied to create a specific coordinate system. Many projections such as various national grids are simply specific parameterized variants of a general projection that has been given a local name. A specific parameterized variant of a coordinate system in Manifold is called a preset.

Although fundamentally new types of coordinate systems cannot be added since these involve the creation of new program logic, users can add new variations of coordinate systems by providing named presets for customized versions of the standard projections within Manifold that do accept parameters.

The definitions of coordinate systems available to Manifold are loaded from .xml files found in the Config folder. If desired, users can add their own variations to incorporate “new projections” into Manifold. An example of a new projection preset is:

```
<xml>
  <preset>
    <name>My System</name>
    <category>National Grids*My Country</category>
    <system>Cassini</system>
    <centerLat>31.73</centerLat>
    <centerLon>35.21</centerLon>
    <falseEasting>170000</falseEasting>
    <falseNorthing>126800</falseNorthing>
  </preset>
</xml>
```

The <system> attribute specifies the base coordinate system that is being customized. Different coordinate systems allow different attributes. The above example shows attributes that can be used with the Cassini coordinate system.

**Rules**

- There can be any number of `<preset>` ... `</preset>` entries within a single XML file between the `<xml>` tag at the beginning of the file and the `</xml>` tag at the end of the file.
- The name string supplied in the `<name>` attribute is mandatory and must be unique.
- `<category>` is mandatory and represents the hierarchical path to the preset within the Projections dialog projections pane. Different levels of the hierarchical path are separated by an asterisk. For example, the preset above will appear within the National Grids folder in a My Country folder.
- `<system>` is mandatory and is the name of an existing Manifold coordinate system.
- Other attributes are parameters specific to the coordinate system, as listed below.

To utilize newly created or modified presets place the new or modified .xml file containing the presets into the Config folder for Manifold (normally C:\Program Files\Manifold System Professional\Config) and restart Manifold.

When Manifold launches, the system will scan all .xml files in the Config folder. Any `<preset>` ... `</preset>` entries found in any of those .xml files will be loaded into the system as available presets.

**Note:** in the tables listing tags below, such as those that may be used for parameters, it is understood that a `<tag>` is matched with a `</tag>`. Instead of writing `<tag>` ... `</tag>` in the tables to indicate a `<tag>` followed by some intermediate material and then closed with a `</tag>`, we simply write “<tag>”

Coordinate System Parameters Available in all Systems

All coordinate systems may include the following optional parameters. These may be used to specify an ellipsoid and other parameters within the coordinate system. See the Custom Ellipsoids and Custom Datums topics for information on ellipsoid and datum specification attributes.
Note: Although it is possible to define a coordinate system using `<majorAxis>`, `<eccentricity>`, `<centerX>`, `<centerY>` and `<centerZ>` tags, users are strongly advised to define a custom datum and then use the `<datum>` tag.

- `<majorAxis>`: Major axis of the ellipsoid.
- `<datum>`: Datum to be used for this coordinate system.
- `<eccentricity>`: Ellipsoid eccentricity.
- `<centerX>`: X offset to datum center in meters.
- `<centerY>`: Y offset to datum center in meters.
- `<centerZ>`: Z offset to datum center in meters.
- `<scaleX>`: X scale correction.
- `<scaleY>`: Y scale correction.
- `<falseEasting>`: False easting in meters or degrees.
- `<falseNorthing>`: False northing in meters or degrees.
- `<localOffsetX>`: X offset in meters or degrees.
- `<localOffsetY>`: Y offset in meters or degrees.
- `<localScaleX>`: X scale in meters or degrees.
- `<localScaleY>`: Y scale in meters or degrees.
- `<rotationX>`: Datum rotation about the X axis, in units specified by the `unit` attribute. By default, the rotation is in degrees.
- `<rotationY>`: Datum rotation about the Y axis, in units specified by the `unit` attribute. By default, the rotation is in degrees.
- `<rotationZ>`: Datum rotation about the Z axis, in units specified by the `unit` attribute. By default, the rotation is in degrees.

The above parameters will be applied in order they occur in the file except that the `<datum>` tag will always be applied before any other tags related to datum information.

Very Important: Note that the false Easting and Northing values are always given in meters or degrees, even in the case of those projections that traditionally use feet. When specifying projection presets one must always convert the desired false Easting and Northing values into meters for use in the XML specification. When the values are viewed in a Manifold coordinate system dialog they are converted to whatever units are active. So, what we see in an XML file is not the same to what we might see in a dialog.

Example

Some parts of South Carolina use a State Plane coordinate system that uses international feet as a measurement unit instead of survey feet. We could add this as a preset using the following XML:

```xml
<xml>
  <preset>
    <name>State Plane - South Carolina, NAD 83, Intl Feet</name>
    <category>Custom Presets</category>
    <system>Lambert Conformal Conic</system>
    <datum>North American 1983 (mean for CONUS)</datum>
    <unit>Foot</unit>
    <centerLat>31.833333333333333333</centerLat>
    <centerLon>-81.0</centerLon>
  </preset>
</xml>
```
Coordinate System Parameters

The following list of valid specifications for coordinate systems allowed in `<system>` attributes includes all parameters that may be specified for each. An attribute is a pair of tags with the value in between, as in `<centerLat>15</centerLat>`. Some coordinate systems, such as Latitude / Longitude have no optional parameters that are specific to that coordinate system. In addition to the parameters specified, all coordinate systems allow specification of common parameters.

Each coordinate system is listed by its `<system>` ... `<system>` name.

<table>
<thead>
<tr>
<th>Coordinate System and Attributes</th>
<th>Description of Attributes (Parameters)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Latitude / Longitude</td>
<td>No parameters.</td>
</tr>
<tr>
<td>Albers Conical Equal Area</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>&lt;centerLat&gt;</code> Center latitude.</td>
</tr>
<tr>
<td></td>
<td><code>&lt;centerLon&gt;</code> Center longitude.</td>
</tr>
<tr>
<td></td>
<td><code>&lt;firstStdLat&gt;</code> First standard latitude.</td>
</tr>
<tr>
<td></td>
<td><code>&lt;secondStdLat&gt;</code> Second standard latitude.</td>
</tr>
<tr>
<td>Azimuthal Equidistant</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>&lt;centerLat&gt;</code> Center latitude.</td>
</tr>
<tr>
<td></td>
<td><code>&lt;centerLon&gt;</code> Center longitude.</td>
</tr>
<tr>
<td>Azimuthal Equidistant (Guam)</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>&lt;centerLat&gt;</code> Center latitude.</td>
</tr>
<tr>
<td></td>
<td><code>&lt;centerLon&gt;</code> Center longitude.</td>
</tr>
<tr>
<td>Bipolar Oblique Conformal Conic</td>
<td>No parameters.</td>
</tr>
<tr>
<td>Bonne</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>&lt;centerLat&gt;</code> Center latitude.</td>
</tr>
<tr>
<td></td>
<td><code>&lt;centerLon&gt;</code> Center longitude.</td>
</tr>
<tr>
<td>Cassini</td>
<td></td>
</tr>
<tr>
<td>Projection</td>
<td>Parameters</td>
</tr>
<tr>
<td>----------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>Cylindrical Equal Area</td>
<td>(&lt;centerLat&gt;, &lt;centerLon&gt;)</td>
</tr>
<tr>
<td>Cylindrical Equidistant</td>
<td>(&lt;centerLat&gt;, &lt;centerLon&gt;)</td>
</tr>
<tr>
<td>Donald</td>
<td>No parameters</td>
</tr>
<tr>
<td>Gall</td>
<td>(&lt;centerLon&gt;)</td>
</tr>
<tr>
<td>Miller Cylindrical</td>
<td>(&lt;centerLon&gt;)</td>
</tr>
<tr>
<td>Eckert IV</td>
<td>(&lt;centerLon&gt;)</td>
</tr>
<tr>
<td>Eckert VI</td>
<td>(&lt;centerLon&gt;)</td>
</tr>
<tr>
<td>Equidistant Conic</td>
<td>(&lt;centerLat&gt;, &lt;centerLon&gt;, &lt;firstStdLat&gt;, &lt;secondStdLat&gt;)</td>
</tr>
<tr>
<td>Gnomonic</td>
<td>(&lt;centerLat&gt;, &lt;centerLon&gt;)</td>
</tr>
<tr>
<td>Goode's Homolosine (Interrupted)</td>
<td>No parameters</td>
</tr>
</tbody>
</table>
Manifold® System Release 8.00 User Manual

Guyou

<centerLon> Center longitude.

Hammer

<centerLon> Center longitude.

Krovak Oblique Conformal Conic

<centerLat> Center latitude.
<centerLon> Center longitude.
<centerLineAzimuth> Azimuth of the center line.
<pseudoStdLat> Pseudostandard latitude.

Lambert Azimuthal Equal Area

<centerLat> Center latitude.
<centerLon> Center longitude.

Lambert Conformal Conic

<centerLat> Center latitude.
<centerLon> Center longitude.
<firstStdLat> First standard latitude.
<secondStdLat> Second standard latitude.

Lambert Conformal Conic (Single Parallel)

<centerLat> Center latitude.
<centerLon> Center longitude.

Mercator

<centerLat> Center latitude.
<centerLon> Center longitude.

Modified Polyconic

<centerLon> Center longitude.
<firstStdLat> First standard latitude.
<secondStdLat> Second standard latitude.
<trueScaleLon> True scale longitude.
Modified Stereographic

<centerLat> Center latitude.
<centerLon> Center longitude.

Modified Transverse Mercator (Only used for Alaska with predefined parameters)

Mollweide

<centerLon> Center longitude.

Mollweide Interrupted No parameters.

New Zealand

<centerLat> Center latitude.
<centerLon> Center longitude.

Oblated Equal Area

<centerLat> Center latitude.
<centerLon> Center longitude.
<ovalRotationAngle> Oval rotation angle.
<ovalShapeM> Oval shape M.
<ovalShapeN> Oval shape N.

Oblique Mercator (A)

<centerLat> Center latitude.
<firstLat> Latitude of the first reference location.
<firstLon> Longitude of the first reference location.
<secondLat> Latitude of the second reference location.
<secondLon> Longitude of the second reference location.

Oblique Mercator (A, centered)

<centerLat> Center latitude.
<firstLat> Latitude of the first reference location.
<firstLon> Longitude of the first reference location.
<secondLat> Latitude of the second reference location.
<secondLon> Longitude of the second reference location.
Oblique Mercator (B)

- `<centerLat>` Center latitude.
- `<centerLon>` Center longitude.
- `<centerLineAzimuth>` Azimuth of the center line.
- `<rectifiedGridAngle>` Rectified grid angle.

Oblique Mercator (B, centered)

- `<centerLat>` Center latitude.
- `<centerLon>` Center longitude.
- `<centerLineAzimuth>` Azimuth of the center line.
- `<rectifiedGridAngle>` Rectified grid angle.

Orthographic

- `<centerLat>` Center latitude.
- `<centerLon>` Center longitude.

Peirce

- `<centerLat>` Center latitude.
- `<centerLon>` Center longitude.

Polyconic

- `<centerLat>` Center latitude.
- `<centerLon>` Center longitude.

Robinson

- `<centerLon>` Center longitude.

Sinusoidal

- `<centerLon>` Center longitude.

Space Oblique Mercator

- `<ascendingNodeLon>` Longitude of the ascending node.
- `<completedOrbits>` Number of fully completed orbits.
- `<orbitInclinationAngle>` Orbit inclination angle.
- `<rotationToAscendingNode>` Earth rotation to the ascending node.
Customization

Space Oblique Mercator for Landsat

- `<orbitNumber>`: LANDSAT ID for the orbit.
- `<orbitPathNumber>`: LANDSAT ID for the orbit path.

Stereographic

- `<centerLat>`: Center latitude.
- `<centerLon>`: Center longitude.

Double Stereographic

- `<centerLat>`: Center latitude.
- `<centerLon>`: Center longitude.

Polar Stereographic

- `<centerLon>`: Center longitude.
- `<trueScaleLat>`: True scale latitude.

Tilted Perspective

- `<centerLat>`: Center latitude.
- `<centerLon>`: Center longitude.
- `<centerHeight>`: Center height in meters.
- `<tiltAngle>`: Tilt angle.
- `<tiltAzimuth>`: Tilt azimuth.

Transverse Mercator

- `<centerLat>`: Center latitude.
- `<centerLon>`: Center longitude.

Van der Grinten

- `<centerLon>`: Center longitude.

Van der Grinten IV

- `<centerLon>`: Center longitude.

Vertical Perspective
<centerLat> Center latitude.
<centerLon> Center longitude.
<centerHeight> Center height in meters.

Wagner IV
<centerLon> Center longitude.

Wagner VII
<centerLon> Center longitude.

Winkel Tripel
<stdLat> Standard latitude

**Tech Tip**

Manifold cannot use any customizations if the .xml files do not contain XML that is exactly correct. A useful test before launching Manifold is to open any newly created or modified .xml file in Internet Explorer. Internet Explorer will show a correct .xml file in a simple text format. If Internet Explorer cannot parse the .xml file, Manifold won't be able to either.

**Caution**

Defining custom presets for coordinate systems is an expert-level task that requires expertise in details of coordinate systems and projections. Correct choice of and use of parameters requires extensive knowledge of projections. Providing an incorrect specification or inappropriate parameters can provide erroneous system function. This capability is included within Manifold for the use of experts. It is not supported by any Manifold technical support product.

**See Also**

The Earth as an Ellipsoid
Projections
Custom Layout Templates

Manifold allows definition of custom layout templates. Layout templates are defined using XML files that contain one or more template definitions.

A layout template is defined using a <layout> tag. A layout template might also have an <elements> tag with one or more definitions for layout elements.

Layout templates are defined by .xml files in the following pattern:

```xml
<xml>
  <layout>
    <name>My Template</name>
    <pagesByX>1</pagesByX>
    <pagesByY>1</pagesByY>
    <elements>
      (layout elements go here)
    </elements>
  </layout>
</xml>
```

Rules

There can be any number of <layout>...</layout> entries within a single XML file between the <xml> tag at the beginning of the file and the </xml> tag at the end of the file. Within <layout> entries the following tags are used:

- <name> is mandatory and must be unique.
- <pagesByX> is optional and specifies the horizontal extent of the layout in pages. If the tag is omitted, applying the template preserves the horizontal extent of the target layout.
- <pagesByY> is optional and specifies the vertical extent of the layout in pages. If the tag is omitted, applying the template preserves the vertical extent of the target layout.
- <elements> is optional and contains zero or more definitions for the layout elements.

Layout Elements

Layout elements that may be defined inside the <elements> tag are:

- <body ref=... back=... background=... backgroundFore=... border=... borderDegMinSec=... borderEachPages=... borderFont=... borderFore=... borderMargin=... borderNoOverlaps=... borderRounding=... borderStep=... borderUnit=... columnHeaders=... controlPoints=... font=... fore=... graticule=... grid=... gridLines=... height=... legend=... northArrow=... pageFilters=... paging=... rotation=... scaleBar=... scope=... scopeDetail=... shadeAlternate=... visible=... width=... xmax=... xmin=... ymax=... ymin=... /> - The <body> tag defines a layout element for the parent component of the layout. If the layout does not have a parent component, the element defined by the <body> tag is ignored. The ref attribute of the <body> tag defines the reference name for the layout element and can be used to bind other layout elements to this element.
- <component ref=... back=... background=... backgroundFore=... border=... borderDegMinSec=... borderEachPages=... borderFont=... borderFore=... borderMargin=... borderNoOverlaps=... borderRounding=... borderStep=... borderUnit=... columnHeaders=... controlPoints=... font=... fore=... graticule=... grid=... gridLines=... height=... legend=... name=... northArrow=... pageFilters=... paging=... rotation=... scaleBar=... scope=... scopeDetail=... shadeAlternate=... visible=... width=... xmax=... xmin=... ymax=... ymin=... /> - The <component> tag defines a layout element for the component the name of which is given by the name attribute. If the project does not contain a component with such a name, or if the name refers to a component that can not participate in a layout, the element defined by the <component> tag is ignored. The ref attribute of the <component> tag defines the reference name for the layout element and can be used to bind other layout elements to this element.
- <legend refParent=... height=... rotation=... visible=... width=... xmax=... xmin=... ymax=... ymin=... /> - The <legend> tag defines a layout element for a legend. The legend is taken from the component used by the layout element with the value of the ref attribute equal to the value of the refParent attribute in the <legend> tag.
Attributes used within the layout elements listed above include the following attributes:

- `back=...` - Specifies the background color of a layout element. See the Custom Palettes and Themes topic for the format of a color value.
- `backgroundColor=...` - Controls the background of a layout element for a mappable component. Can be set to one of the following values: "auto" (turned on or off according to a component), "hide" (turned off) and "custom" (turned on, uses a custom color).
- `backgroundColorFore=...` - Specifies the custom background color of a layout element for a mappable component. If the value of the background attribute is not "custom", the value of the `backgroundColor` attribute is ignored.
- `border=...` - Specifies the border style of a layout element for a mappable component. Can be set to one of the following values: "none" (turned off), "coordinates" (turned on, uses custom coordinates), "coordinates (graticule)" (turned on, uses graticule coordinates), "coordinates (grid)" (turned on, uses grid coordinates), "thick" (turned on, thick) or "thin" (turned on).
- `borderDegMinSec=...` - Controls the format of lat/lon border coordinates of a layout element for a mappable component. Can be set to one of the following values: "true" (lat/lon coordinates use the degree-minute-second format) or "false" (lat/lon coordinates use the decimal degrees format).
- `borderEachPage=...` - Controls the appearance of the border of a layout element for a mappable component in a multipage layout. Can be set to one of the following values: "true" (the border is replicated on each page) or "false" (the border is shared between pages).
- `borderFont=...` - Specifies the font used to format the border coordinates of a layout element for a mappable component. See the notes at the end of this list for the format of the font value.
- `borderFore=...` - Specifies the color of the border of a layout element for a mappable component. See the Custom Palettes and Themes topic for the format of a color value.
- `borderMargin=...` - Specifies the width in printer's points of the border margin of a layout element for a mappable component.
- `borderNoOverlaps=...` - Turns clipping of overlapping labels in the border of a layout element for a mappable component on and off. Can be set to one of the following values: "true" (overlapping labels are clipped) or "false" (overlapping labels are displayed on top of each other).
- `borderRounding=...` - Specifies the number of digits to use when rounding the coordinate values in the border of a layout element for a mappable component. A value of 1 rounds to 1/10ths, a value of 0 rounds to whole numbers, a value of -1 rounds to tens and so on.
- `borderStep=...` - Specifies the interval between the coordinate values in the border of a layout element for a mappable component. The units of the interval are given by the value of the `borderUnit` tag. If the value of the border attribute is not "coordinates", the value of the `borderStep` attribute is ignored.
- `borderUnits=...` - Specifies the unit to use for the coordinate values in the border of a layout element for a mappable component. The number of units in the interval is given by the value of the `borderStep` tag. If the value of the border attribute is not "coordinates", the value of the `borderUnits` attribute is ignored.
- `columnHeaders=...` - Turns column headers in a layout element for a table component on and off. Can be set to one of the following values: "true" (turned on) or "false" (turned off).
• `controlPoints` - Specifies the output of the control points in a layout element for a mappable component. Can be set to one of the following values: "auto" (turned on or off according to a component), "hide" (turned off) and "show" (turned on).

• `font` - Specifies the font to use for a layout element for a comment, query, script or table component. See the notes at the end of this list for the format of the font value.

• `fore` - Specifies the foreground color of a layout element. See the Custom Palettes and Themes topic for the format of a color value.

• `graticule` - Controls the output of the graticule in a layout element for a mappable component. Can be set to one of the following values: "auto" (turned on or off according to a component), "hide" (turned off) and "show" (turned on).

• `grid` - Controls the output of the grid in a layout element for a mappable component. Can be set to one of the following values: "auto" (turned on or off according to a component), "hide" (turned off) and "show" (turned on).

• `gridLines` - Turns grid lines in a layout element for a table component on and off. Can be set to one of the following values: "true" (turned on) or "false" (turned off).

• `height` - Specifies the height of a layout element. See the notes at the end of this list for the format of the height value. If the layout element specifies both the `ymin` and `ymax` attributes, the value of the height attribute is ignored.

• `legend` - Controls the output of the legend in a layout element for a mappable component. Can be set to one of the following values: "auto" (turned on or off according to a component), "hide" (turned off) and "show" (turned on).

• `name` - Specifies the name of a component for a layout element defined by the `<component>` tag.

• `northArrow` - Controls the output of the north arrow in a layout element for a mappable component. Can be set to one of the following values: "auto" (turned on or off according to a component), "hide" (turned off) and "show" (turned on).

• `pageFilter` - Specifies the page filter for a layout element in a multipage layout.

• `paging` - Specifies the paging mode for a layout element in a multipage layout. Can be set to one of the following values: "continuous" (the component is stretched across all pages) or "individual" (the component is reproduced similarly on every page).

• `ref` - Specifies the reference name for a layout element defined by a `<body>` or `<component>` tag.

• `refParent` - Specifies the reference name of a parent layout element for a layout element defined by a `<legend>`, `<northArrow>`, `<scaleBar>` or `<text>` tag.

• `rotation` - Specifies the rotation angle of a layout element, in degrees.

• `scaleBar` - Controls the output of the scale bar in a layout element for a mappable component. Can be set to one of the following values: "auto" (turned on or off according to a component), "hide" (turned off) and "show" (turned on).

• `scope` - Specifies the scope of a layout element for a mappable component. Can be set to one of the following values: "entire component" (entire component), "locked rectangle" (a rectangle specified by the value of the `scopeDetails` attribute), "locked center / scale" (a view specified by the value of the `scopeDetails` attribute), "layer" (a layer specified by the value of the `scopeDetails` attribute), "selection" (the selection), "saved selection" (a saved selection specified by the value of the `scopeDetails` attribute), "view" (a view specified by the value of the `scopeDetails` attribute).

• `scopeDetail` - Specifies additional data for the scope of a layout element for a mappable component. If the value of the `scope` attribute is "locked rectangle", the value of the `scopeDetail` attribute is in the form "xmin,ymin,xmax,ymax", where `xmin`, `ymin`, `xmax` and `ymax` are the coordinates of a rectangle in the coordinate system of the component. If the value of the `scope` attribute is "locked center / scale", the value of the `scopeDetail` attribute is in the form "xcenter,ycenter, scale", where `xcenter` and `ycenter` are the center coordinates of a view in the coordinate system of the component and `scale` is the number of the coordinate units in the coordinate system of the component per pixel. If the value of the `scope` attribute is "layer", "saved selection" or "view", the value of the `scopeDetail` attribute is the name of the respective layer, saved selection or view.

• `shadeAlternate` - Turns shading of alternate rows in a layout element for a table component on and off. Can be set to one of the following values: "true" (turned on) or "false" (turned off).

• `size` - Specifies the width of the line for a layout element for a line or the width of the border line for a layout element for a rectangle. Can be an integer or one of the following values: "1/2", "1/3", "1/4", "1/5", "1/6", "1/8", "1/10", "1/12", "1/16", "1/20".

• `text` - Specifies the text of a text layout element. The text can include one or more escape sequences.

• `textAlignX` - Specifies the horizontal alignment of a layout element for text. Can be set to one of the following values: "center", "left", "right", "justify".

• `textAlignY` - Specifies the vertical alignment of a layout element for text. Can be set to one of the following values: "center", "top", "bottom".
• `textDecimals=...` - Specifies the number of decimal digits to use for escape sequences in a layout element for text.
• `visible=...` - Turns a layout element on and off. Can be set to one of the following values: “true” (visible) or “false” (hidden).
• `width=...` - Specifies the width of a layout element. See the notes at the end of this list for the format of the width value. If the layout element specifies both the `xmin` and `xmax` attributes, the value of the width attribute is ignored.
• `xmax=...` - Specifies the right bound of a layout element. See the notes at the end of this list for the format of the bound value.
• `xmin=...` - Specifies the left bound of a layout element. See the notes at the end of this list for the format of the bound value.
• `ymax=...` - Specifies the top bound of a layout element. See the notes at the end of this list for the format of the bound value.
• `ymin=...` - Specifies the bottom bound of a layout element. See the notes at the end of this list for the format of the bound value.

### Specifying Fonts, Bounds and Dimensions

Fonts are specified using the following format:

```plaintext
facename, size[, bold][, italic][, strikeout][, underline]
```

...for example,

```
Verdana, 8
Arial, 10, bold
Lucida Sans Unicode, 9, bold, italic
```

Bounds and dimensions of layout elements are specified as:

- A fraction of the page, for example, "0.5" for half of the page, "1.0" for the entire page.
- A percentage of the page, for example, "50%" for half of the page.
- A value in physical units, for example, "1in" or "1 in" or "1 inch" for 1 inch.

### See Also

- [Layouts](#)
- [Layout Templates](#)
- [Custom Palettes and Themes](#)
Command Filters

When configuring Manifold for use by naïve users an administrator may want to simplify the command interface presented by Manifold to eliminate commands that a naïve user should not be allowed to use. For example, we may want to eliminate commands such as the Edit - Projection command that can be used to change projections. This is easily accomplished using command filters.

A command filter is an XML file containing an arbitrary number of `<enable>` and `<disable>` entries. Each entry contains a regular expression pattern. Manifold uses these entries to enable or disable commands within the Manifold menu structure.

Launching Manifold with the `/cfilter:<path>` option starts Manifold using a command filter. For example, entering the following at the Windows command prompt (within the installation folder for Manifold) or within a batch file will launch Manifold using a command filter:

```bash
manifold.exe /cfilter:\myfiles\customizations\mymanifold.xml
```

The above will use the `mymanifold.xml` command filter file located in the `\myfiles\customizations\` folder.

To decide if a given command is enabled or disabled by the filter file, Manifold traverses all entries in the order they appear in the file, and matches the command name against the pattern in the entry. Case is significant in command names. If the command name matches the pattern, Manifold enables or disables the command as specified.

Example

The following command filter disables the Tools - Options command:

```xml
<xml>
  <disable>ToolsOptions</disable>
</xml>
```

Example

The following command filter disables all commands in the Tools menu, except the Tools - Options command:

```xml
<xml>
  <disable>Tools.*</disable>
  <enable>ToolsOptions</enable>
</xml>
```

Example

The following command filter disables all commands in the Tools and Drawings menus:

```xml
<xml>
  <disable>Tools.*</disable>
  <disable>Drawing.*</disable>
</xml>
```

Discovering Command Names

Launching Manifold with the `/clist:<path>` option creates a list of all available command names. For example, from the Windows command prompt (within the installation folder for Manifold) entering:

```bash
manifold.exe /clist:\commandlist.txt
```
…will create a `commandlist.txt` file in the root with a long list of command names, such as:

```plaintext
ChartData
ChartDataMoveDown
ChartDataMoveToBottom
ChartDataMoveToTop
ChartDataMoveUp
ChartDataSelectAll
ChartDataSelectInverse
ChartDataSelectNone
ChartDataShowTypes
ChartFormat
```

…and so on.

This provides a comprehensive list of names to use for command names when creating command filters. Keep in mind that some listed commands may not be available because of current activation status. For example, if Surface Tools have not been activated then any commands within Surface Tools cannot be enabled.

In addition, the list created by `clist` will include all internal commands known to Manifold, some of which are names that are reserved for future extensions.
Add-Ins

Add-Ins have two functions:

- Add-ins add new commands to the Custom toolbar or to the Tools - Add-Ins menu. Some of these commands may invoke custom panes.
- Add-ins control which Manifold toolbars are shown, hidden or disabled when the system starts up. Only one of the installed add-ins can control toolbars. To select which add-in controls toolbars, use the Add-In Manager dialog.

Add-ins are defined using .xml files that are placed within the Manifold System Config folder. To add new commands a text file containing a Manifold Script will also be required. Any Manifold script can be used, except those that include forms. Add-ins cannot execute scripts involving forms. When an Add-in executes, it always executes in non-debug mode even if the Debugger has been installed. To debug scripts, run them as ordinary scripts within the project using the debugger and then after they are debugged place them in a text file for use by the Add-in.

Add-ins may be written in any ActiveX scripting language or .NET language that is available on the machine.

Add-ins are defined by .xml files in the following pattern:

```xml
<xml>
  <addin>
    <name>My Add-in Name</name>
    <copyright>Copyright MyCompany Inc., 2001.</copyright>
    <description>Description goes here</description>
    <command>
      <icon>filename.png</icon>
      <name>Name to put in menu</name>
      <path>scriptfile.txt</path>
      <scriptEngine>VBScript</scriptEngine>
      <status>This text appears on the status bar</status>
      <tip>Tool tip text</tip>
    </command>
    <toolBar>
      <name>Toolbar Name</name>
      <display>hide</display>
    </toolBar>
  </addin>
</xml>
```

Rules

There can be any number of `<addin>` ... `<addin>` entries within a single XML file between the `<xml>` tag at the beginning of the file and the `</xml>` tag at the end of the file. Within `<addin>` entries the following attributes are used:

- `<name>` is mandatory and must be unique.
- `<command>` is optional. A `<command>` entry defines a command associated with the add-in.
- `<copyright>` is optional. It is a copyright string displayed within the Add-In Manager dialog.
- `<description>` is optional. It should be a short string that is displayed within the Add-In Manager dialog.
- `<reference>` is optional and contains a reference to an external module for .NET scripts. There can be more than one `<reference>` tag in the `<addin>` tag.
There can be any number of `<command>` ... </command> entries within an `<addin>` entry. Within `<command>` entries the following attributes are used:

- `<toolbar>` is optional. A `<toolbar>` entry controls the startup configuration of a Manifold System toolbar.

- `<icon>` contains a path to a graphics file containing the icon to be used for the toolbar button. The path can be either absolute or relative to the location of the Config folder. Graphics files can be in the same formats supported for icon point styles (BMP, GIF, JPG, PNG, TIF, etc) and will be autoscaled to 16 pixels wide and 15 pixels high in size. If the `<icon>` tag is present Manifold will create a new toolbar button on the Custom toolbar for this add-in. The `<icon>` tag may be omitted if the `<name>` tag is present. If there is no `<icon>` tag there will be no button created on the Custom toolbar for this add-in.

- `<form>` contains the programmatic name of the form to use for the command in a menu. If the form is an ActiveX control, the `<form>` tag should contain the programmatic ID of the control object. If the form is a .NET control, the `<form>` tag should contain either a name or a path to the assembly containing the control class and the name of the class, separated by a semicolon. The path to the assembly can be either absolute or relative to the location of the Config folder. If the `<form>` tag is present, the `<path>` tag can be omitted. If both the `<form>` and the `<path>` tags are present, the `<form>` tag is ignored.

- `<name>` contains the name string to use for the command in a menu. If the `<name>` tag is present, Manifold will create a new entry with that name for this add-in in the Tools - Add-Ins menu. If present, the `<name>` text must be unique. The `<name>` tag may be omitted if the `<icon>` tag is present. If there is no `<name>` tag within a `<command>` there will be no menu entry created.

- `<path>` contains the path to a text file containing the script text for the command. The path can be either absolute or relative to the location of the Config folder. If the `<form>` tag is present, the `<path>` tag can be omitted, otherwise the `<path>` tag is mandatory. If both the `<form>` and the `<path>` tags are present, the `<form>` tag is ignored.

- `<scriptEngine>` is optional and contains the programmatic ID of the ActiveX script engine or a string literal identifying the .NET script engine (".NET", "JScript.NET", "IronPython.NET", or "VB.NET"). If omitted, the system assumes the add-in is written in VBScript.

- `<nameStrict>` is optional and controls name recognition. Add-in commands can use the `<nameStrict>` tag to specify strict command names for filtering purposes. The system makes sure all add-in commands start with "Addin", so a command named "AddinCommand" is registered as "AddinCommand", but a command named "Tool" is registered as "AddinTool".

- `<status>` is optional and provides text to place on the main Manifold status bar when the mouse hovers over the toolbar button associated with the add-in command.

- `<tip>` is optional and provides tool tip text that appears when the mouse hovers over the toolbar button associated with the add-in command.

There can be any number of `<toolbar>` ... </toolbar> entries within an `<addin>` entry. Within `<toolbar>` entries the following attributes are used:

- `<name>` is mandatory and contains the toolbar name as it is shown in Manifold.

- `<display>` is mandatory and specifies the toolbar display state as show, hide or disable.

In the `<display>` attribute, show means to show the toolbar by default for those components that normally use that toolbar. hide means to not display the toolbar by default but to allow it to be chosen for display through the Tools - Customize dialog. disable does not show the toolbar and also prevents it from being listed in the Tools - Customize dialog. Use disable to prevent a toolbar from ever being used.

Adding Commands

Adding a command requires writing a script that implements the command and then arranging an Add-in that adds the command to the system when Manifold is launched. Write the script text into a .txt file, create a graphics file for the toolbar button icon and create an .xml file. Place the files within the Config folder hierarchy for Manifold (normally C:\Program Files\Manifold System\Config) and restart Manifold.

When Manifold launches, the system will scan all .xml files in the Config folder and all sub-folders within the Config folder. Any `<addin>` ... </addin> entries found in any of those .xml files will be loaded into the system as available add-ins.

Example
We will add a toolbar button that when pressed pops open a message box to say "Hello, world!" To do so, we create a subfolder in the Config folder called hello. Within hello we’ve created three files: hello.xml, hi.png and hello.txt.

The hi.png file contains an image that is 16 pixels wide and 15 pixels high.

![Hi.png](attachment:Hi.png)

Seen in Manifold with the border turned on, the hi.png file contains pixels spelling out the word "Hi" in red color. Other pixels are transparent color. In menus, Manifold will replace transparent color with light gray color (192, 192, 192) to match default Windows desktop settings.

The hello.txt file contains the scripting text to use:

```vbs
Sub Main
    Application.MessageBox "Hello, world!", "Traditional Greeting"
End Sub
```

The "Traditional Greeting " text specifies the text to put into the title bar of the message box.

The hello.xml file contains the XML configuration information to add the add-in into Manifold:

```xml
<xml>
    <addin>
        <name>Hello World Add-In</name>
        <copyright>Copyright MyCompany Inc, 2001.</copyright>
        <description>Simple add-in with single command.</description>
        <command>
            <icon>hello\Hi.png</icon>
            <name>Say Hello</name>
            <path>hello\hello.txt</path>
            <scriptEngine>VBScript</scriptEngine>
            <status>Displays traditional first program text.</status>
            <tip>Say Hello</tip>
        </command>
    </addin>
</xml>
```

Note that the paths to the filenames in the `<icon>` and `<path>` attributes are given relative to the Config folder.

When Manifold launches it will see the hello.xml file within the hello folder in the Config folder and will read it to see what it contains. It will find an add-in and will place the icon on the Custom toolbar as well as placing a Say Hello entry within the Tools - Add-Ins menu. When that toolbar button is pressed or the menu command is invoked the script in hello.txt will be executed and a message box will pop open.

This example is a very simple one. Since any Manifold script (except those using a form) can be executed, very complex add-ins can be created. In fact, one could install one or more COM objects and then call those objects from within an add-in.

**Advanced Example: Toolbar Configuration**
Add-ins can control the default configuration of toolbars by including `<toolbar>` entries. In fact, an add-in might not have any `<command>` entries but only have `<toolbar>` entries to control default toolbar configuration without adding any commands.

This example includes both a command as well as toolbar configuration entries. It uses the same `hi.png` graphics file as well as a `hi.txt` script file and a `hi.xml` XML file. All three are placed within a folder called `hi` within the `Config` folder.

The `hi.txt` file contains:

```vbscript
Sub Main
    Application.MessageBox "Hello from advanced add-in!", "Advanced Add-In"
End Sub
```

The `hi.xml` file contains:

```xml
<xml>
<addin>
    <name>Hi Advanced</name>
    <copyright>Copyright MyCompany Inc, 2001.</copyright>
    <description>Advanced add-in with single command.</description>

    <command>
        <icon>Hi\Hi.png</icon>
        <name>Say Hi (Advanced)</name>
        <path>Hi\Hi.txt</path>
        <scriptEngine>VBScript</scriptEngine>
        <status>Displays advanced welcome message.</status>
        <tip>Say Hi (Advanced)</tip>
    </command>

    <toolBar>
        <name>Alignment</name>
        <display>hide</display>
    </toolBar>
    <toolBar>
        <name>Query</name>
        <display>hide</display>
    </toolBar>
    <toolBar>
        <name>Tools</name>
        <display>disable</display>
    </toolBar>
    <toolBar>
        <name>Tools (Advanced)</name>
        <display>disable</display>
    </toolBar>
    <toolBar>
        <name>Tracing</name>
        <display>hide</display>
    </toolBar>
    <toolBar>
        <name>Transform</name>
        <display>hide</display>
</addin>
</xml>
```
The most important thing about the above XML is that it completely disables the **Tools** and **Tools (Advanced)** toolbars that would normally be used to create forms.

**Advanced Example: Creating a Custom Pane**

Add-ins can be used to create custom panes. This example demonstrates how to create a custom pane using Visual Basic 2005 Express.

1. Launch Visual Basic 2005 Express. Create a new Class Library project (File - New Project, Class Library) named **MyPane**.

2. In the Solution Explorer window, right click **Class1.vb** and select **Delete**.

3. In the Solution Explorer window, right click **MyPane** and select **Add - User Control**. Set the control name to **MyPaneControl** and click **Add**.

4. In the **MyPaneControl.vb** design window, add a button (Toolbox, Common Controls, Button) and a label (Toolbox, Common Controls, Label). Double click the button to add a handler for the button's Click event.

5. In the Solution Explorer window, right click **MyPane** and select **Add Reference**. In the **Add Reference** dialog, switch to the **Browse** tab, locate the Manifold installation folder, select **Manifold.Interop.dll** and **Manifold.Interop.Scripts.dll** and click **OK**.

6. In the **MyPaneControl.vb** code window, select all text and replace it with:

```vbnet
Public Class MyPaneControl
    Implements Manifold.Interop.Scripts.IEventsConnection

    Dim app As Manifold.Interop.Application

    Private Sub Button1_Click( _
        ByVal sender As System.Object, _
        ByVal e As System.EventArgs) Handles Button1.Click
        If Not (app Is Nothing) Then
            app.MessageBox("The document contains " + _
                ", component(s).")
        End If
    End Sub

    Public Sub ConnectEvents( _
        ByVal ev As Manifold.Interop.Scripts.Events) _
        Implements Manifold.Interop.Scripts.IEventsConnection.ConnectEvents
        AddHandler ev.DocumentClosed, AddressOf Document_Changed
        AddHandler ev.DocumentCreated, AddressOf Document_Changed
        AddHandler ev.DocumentOpened, AddressOf Document_Changed
        AddHandler ev.DocumentSaved, AddressOf Document_Changed
    End Sub

    Private Sub Document_Changed( _
        ByVal sender As System.Object, _
        app = args.Document.Application
        Label1.Text = args.Document.Path
    End Sub
End Class
```
7. Invoke Project - Show All Files. In the Solution Explorer window, expand MyPane then MyProject and finally double click AssemblyInfo.vb. Locate the line which sets the value of the ComVisible attribute to False, and change it to set the value of the attribute to True:

```
<Assembly: ComVisible(True)>
```

8. Build the class library by using Build - Build MyPane. Make sure there are no build errors.

9. Save the code and the compiled binary by using File - Save All.

10. Launch Windows Explorer and locate the folder you saved the project to. Descend into bin\release, copy MyPane.dll and paste it into the Manifold configuration folder (usually, C:\Program Files\Manifold System\Config). If you do not have write permissions in the configuration folder, launch Manifold and change the file location for that folder to point to where you have write permissions.

11. In Windows Explorer, create a new XML file in the configuration folder and name it MyPane.xml for convenience (this file is different from MyPane.xml generated by Visual Basic). Drag and drop the created MyPane.xml into the opened instance of Visual Basic, set its text to:

```
<?xml version="1.0"?>
<xml>
  <addin>
    <name>My Pane</name>
    <command>
      <name>My Pane</name>
      <form>MyPane.dll;MyPane.MyPaneControl</form>
    </command>
  </addin>
</xml>
```

Save the MyPane.xml file. That's all!

To test the pane, launch a new instance of Manifold. Note that the Tools - Add-Ins menu contains a new command named My Pane. Invoke that command to show the pane. Create a new project. The pane label should become blank. Create a new drawing, then press the pane button. The pane should report that the project .map file contains two components (a drawing and a table). Save the project. The pane label should update its text with the path to the .map file.

**Add-ins and Forms**

Add-ins can include forms, used as modeless panes. Forms can be either ActiveX controls or .NET controls.

To include a form into an add-in, create a command item using the `command` tag, supply the name of the form using the `name` tag and supply either the ProgID of the ActiveX control or the name of the .NET control using the `form` tag. The name of the .NET control should include the name of the assembly or path to the assembly module (either absolute or relative from the Config folder) and the name of the .NET class separated by a semicolon.

**Tech Tips**

Manifold cannot use any customizations if the .xml files do not contain XML that is exactly correct. A useful test before launching Manifold is to open any newly created or modified .xml file in Internet Explorer. Internet Explorer will show a correct .xml file in a simple text format. If Internet Explorer cannot parse the .xml file, Manifold won't be able to either.

Add-in forms implemented as .NET controls can subscribe to events fired by the Manifold core at runtime. To do this, include a reference to Manifold.Interop.dll and Manifold.Interop.Scripts.dll, implement the
The `IEventsConnection` interface from the latter dll and add handlers to the desired events in the `ConnectEvents` method of the interface.

**See Also**

- Tools - Add-In Manager
- User Interface Scripting
- Programming Manifold
Enterprise Edition

Enterprise Edition

Manifold System **Enterprise Edition** is a version of Manifold System that includes additional capabilities not found in the Professional Edition or Personal Edition versions. If you have purchased and installed Enterprise Edition the features described in this and other Enterprise topics will be available to you. If you have installed the standard Professional Edition or Personal Edition of Manifold System you will not be able to use these features.

Enterprise Edition features are also found in Enterprise superset editions of Manifold System, such as Database Administrator Edition or Universal Edition.

In addition to support for classic desktop and server storage models, Manifold's Enterprise Edition also provides a new storage model, a shared Enterprise server model, which combines the advantages of both desktop and server storage. Users new to Manifold sometimes think that using Enterprise Edition requires use exclusively of Enterprise servers. That's not the case: Enterprise Edition supports all three storage models. In fact, all three storage models can be used at the same time within the same project and often are.

Please read the Data Storage Strategies topic to understand how Enterprise servers fit into classic ideas about GIS data storage.

**Enterprise Edition Features**

**Export to .e00 Format** - Enterprise Edition allows export to ESRI .e00 format as well as to the export formats for drawings supported by Professional Edition (ESRI .shp, MapInfo .mid/.mif, Manifold 4.50, AutoCAD .dxf and US Federal SDTS format).

**Shared Storage in Centralized DBMS** - Enterprise Edition can store any Manifold component within a database using a specialized storage model unique to Manifold System called an **Enterprise server**. A Manifold Enterprise server can be hosted on almost any enterprise-class database management system such as IBM DB2, Microsoft SQL Server, and Oracle. DB2, SQL Server and Oracle are officially supported while many other DBMS systems, such as MySQL have been tested and are known to work but are not officially supported.

**Use One Component in Many Projects** - A single copy of a component like a drawing or image can be used in hundreds of projects, but only a single copy need be kept within the Enterprise server. This saves disk space when the same component is used in many different projects.

**Project Coordination via Enterprise Servers** - Enterprise Edition enforces access control and controlled editing of shared components so that many users at once can use the same components. Edits can be made using an easily understood "check out" procedure so that simultaneous, incompatible edits cannot be made. Users can continue using shared components even when someone else has checked them out for editing.

**Concurrent Multi-User Editing** - In addition to the simplified, documented oriented storage model offered by Enterprise Edition using Manifold Enterprise servers, Enterprise Edition also supports truly concurrent, fully multi-user, fully-scalable editing of linked drawings at the object level using drawings linked in a classic server storage model. This allows multi-user editing of drawings of virtually limitless size by a virtually unlimited number of users with storage using standard DBMS geometry types or Oracle Spatial SDO_GEOMETRY that is accepted industry-wide.

**Native Spatial DBMS Support** - Enterprise Edition can store and retrieve vector geometry (drawing objects) and images within native spatial DBMS products such as Oracle Spatial, IBM DB2 with Spatial Extender, SQL Server 2008 spatial (Katmai) and PostgreSQL. Drawings may be exported into spatial DBMS storage with automatic use of the spatial DBMS package's native geometry types, spatial indices and supporting facilities. In addition, Enterprise Edition can store images and surfaces within Oracle databases that include Oracle's GeoRaster technology.

**Generic Spatial Indices** - Enterprise Edition can store drawings, images and surfaces within almost any ordinary DBMS in a way that provides true spatial DBMS capability even if the DBMS does not have its own geometry types and associated spatial facilities such as spatial indices. Once Enterprise Edition has been used to store components in this way then any Manifold license can utilize those components.

**Manifold Spatial Extender for SQL Server** - Enterprise Edition is required to create spatial indices using the Manifold Spatial Extender for SQL Server. Once drawings have been uploaded into SQL Server that is equipped with the Manifold spatial extender, any Manifold edition may be used to import or link those drawings.
Area of Interest Windowing - A key part of unlimited, Enterprise class deployment of GIS applications is the ability to constrain display and editing to a specific, desired region, the Area of Interest (AOI), that is part of what is usually an immensely larger drawing. Manifold Enterprise Edition can work with spatial databases to allow fast and easy specification of an area of interest within a larger drawing: only that part of the drawing will be extracted for use with Manifold, and the part extracted will be fully dynamic and interactive with all viewing, editing and analytic capabilities available to any linked drawings. This Enterprise Edition feature makes it practical for potentially thousands of users to each work with their own areas of interest within drawings that could be terabytes in size. Enterprise Edition is required to use AOI windowing with native spatial DBMS and is required to set up the spatial indices that provide AOI windowing in non-native spatial DBMS.

Manifold Enterprise Server Shared Storage Model

The Enterprise server storage model is an innovation by Manifold that is unique to Manifold System. As discussed in the Data Storage Strategies topic, it provides a storage model that combines the simplicity and convenience of the desktop, document-oriented storage model with many of the organizational benefits of the more complex server, object-oriented storage model while in general avoiding the need for managers to become expert DBMS administrators. It is a tremendous convenience for organizations or individual users who have large or complex data holdings.

Centralized storage via Enterprise servers allows multi-user sharing of common components so different users can simultaneously use the same component within their projects. For example, many users in a town's GIS department can use the same base map drawing of a town within their projects. That means any changes to that base map drawing need be made only in one place. Another benefit to storage of components within an Enterprise server, in addition to the shared access thus provided, is the ability to share a single copy of a component within many different projects so that room for only one copy of the component need be consumed on disk.

For example, if the same drawing of Europe or the US is frequently re-used as a background layer in a map we can share that drawing into an Enterprise server and then share it into a project whenever needed. Only one copy of the drawing will be maintained, the copy within the Enterprise server. If a drawing or other component (such as an image) is used in hundreds of different projects we will see considerable savings in disk space.

A tremendous advantage of Enterprise server storage for both organizations and individuals comes when we need to update some component that is used in many projects. Let's consider an example:

Suppose we have a drawing, such as a boundary lines layer, that is often used as a background layer in maps we create. Over time such a drawing might be used in hundreds of projects that are mixed in among thousands of projects a particular consultant or organization might create. Suppose one day that drawing has to be updated with some changes to the boundaries. That can be a really difficult job if ordinary, desktop computing document storage is used to save everything in a project file.

This is a nightmarish situation many GIS professionals have encountered, when they discover that they have to figure out which of many thousands of project files that might be scattered across disk drives on many machines use that particular drawing. Worse yet, to update all of those projects we would have to open each individual project file and then copy the updated layer from a new project and paste it into that project to update it. What a hassle!

In contrast, if we used Enterprise Edition we could have kept the boundaries layer on an Enterprise server and then linked it into every project that used it as a shared component. In that case, when it came time to update the boundaries layer all we would need to do is simply check it out from the Enterprise server, edit it and then check it back in. When we check it back in, every project that uses that drawing no matter where it is located will be automatically updated. In this case, using Enterprise Edition changes a nightmarish project that we never could be certain would be accomplished even though weeks of effort might go by into a very simple project that could be finished in a few minutes with total certainty that it had been done right.

Obviously, the sooner we start using Enterprise servers to store our data the sooner we can benefit from centralization of our data holdings. Many experienced individuals will begin storing their data in Enterprise servers even when the number of drawings they use and the projects they create are relatively few. They know that in the blink of an eye what starts out as small holdings will grow into many gigabytes of components and projects.

Database Servers
Database vendor websites provide free downloads of complete installation packages for the Express editions of all three of the "Big 3" enterprise-class database management systems supported by Manifold Technical Support for use with Enterprise Edition. Installing any of these three database packages gives Manifold users the ability to achieve high performance, centralized DBMS storage within Manifold Enterprise servers using world-class DBMS products without any additional purchase required.

In alphabetical order, DBMS packages supported by Manifold include:

- **IBM DB2 Express-C** - Based on the IBM DB2 engine. DB2 Express-C is limited to 4 GB of RAM and two processors, but has no limits on database size or the number of users. DB2 provides the largest capacity of any of the Express editions. See the IBM DB2 Express-C Edition topic for more. Native spatial DBMS capability is available for DB2 via the IBM Spatial Extender for DB2, which is a free download for DB2 Express-C users.

- **Oracle 10g Express Edition** - Based on the Oracle 10g Enterprise engine. Oracle 10g Express is limited to 4GB maximum database size, up to 1 GB of RAM and one processor core. Oracle Express includes Oracle's famous spatial technology for high performance storage of drawing geometry, enabling Manifold area of interest windowing. See the Oracle Express Edition topic for more.

- **SQL Server 2005 and 2008 Express** - Based on the SQL Server engine. SQL Server 2005 Express and 2008 Express are limited to 4GB maximum database size, up to 1 GB of RAM and one processor core. SQL Server includes exquisite integration with Microsoft development environments. See the SQL Server Express Edition topic for more. SQL Server 2005 Express (SP2 and greater) may be used as a spatial DBMS when the Manifold Spatial Extender for SQL Server has been installed on the SQL Server machine.

**Shared Storage on Enterprise Servers**

In addition to support for standard desktop storage models and standard server storage models, Manifold's Enterprise Edition uses a new approach to centralized storage of data within DBMS providers.

The standard server storage model as used by classic spatial DBMS is to store objects within a DBMS at the "atomic" object level. Although there are many merits to this approach and it is fully supported by the extensive spatial DBMS capabilities of Manifold, it can be overkill for many users. It can place a greater load on the DBMS administrator, requiring a higher level of technical expertise to keep things running smoothly.

Enterprise Edition provides an alternative, which combines the conveniences of desktop storage with the organizational strengths of server storage. Shared storage using Manifold Enterprise servers stores the entire component within the DBMS, basically using the DBMS as a centralized file cabinet with supervisory capabilities.

In the Manifold shared Enterprise server model, when a drawing, image, surface or other component is shared on a database the master copy of the data is kept within the DBMS, called an Enterprise server. Whenever a user wants to work with a component saved in the Enterprise server, he or she can link the component into the project at hand. Behind the scenes, Manifold fetches a temporary working copy of the entire component into local memory.
The Manifold model has many benefits, including excellent speed once remote components are linked into the local project as compared to high-granularity older architectures. All operations happen within local memory so operations using shared components are just as fast as those using local ones. If a user wishes to edit a shared component, he or she can check it out for local editing, and then check it back into the Enterprise server when done editing.

This check out / check in model of usage is inspired by the highly successful source code control model used within many development environments for collaborative program development by large teams. It is well understood in the software industry and has proven very efficient as a means of managing multiple edit operations by many users.

**Note:** In all fairness, there are faster ways of running classic server storage models than are used by some legacy systems. For example, Manifold Enterprise Edition can connect to Oracle Spatial or other data stores using classic server model linked drawing storage. However, Manifold knows to use intelligent caching whenever possible so that more efficient DBMS accesses can be commanded for much faster performance than possible with some old fashioned systems.

When Manifold Enterprise Edition is used with linked drawings from a fast DBMS data source such as DB2, Oracle, or SQL Server the performance can approach or exceed that of local storage. In some cases, such as area of interest windowing into very large data sets stored on Oracle spatial servers, the performance will significantly exceed that of either local storage or shared Enterprise server storage.

**Advantages of Shared Enterprise Server Storage**

Enterprise Edition storage within Enterprise server geospatial databases has many benefits compared to either desktop models or classic server storage models. The shared Enterprise server installation provides the following benefits:

- Faster performance, at times by a factor of 100 or more in interactive GIS operations as compared to older architectures that have high granularity.
- Reliable, conflict-free editing of complex data in an environment where many users share the same data. Surprisingly, even some very expensive older systems cannot guarantee conflict-free editing as does Manifold.
- Freedom to choose almost any DBMS vendor. Enterprise Edition works with dozens of different database management systems without any need for special DBMS versions, special modules or costly middleware. The Manifold model by default allows use of almost any DBMS that allows transactions and that has an ODBC or OLE DB driver.
- Heterogeneous, free form utilization of Enterprise servers. Within the same project Manifold can transparently include multiple components from different Enterprise servers hosted by different DBMS vendors.
- DBMS safety. If the DBMS crashes while working on an Enterprise project the project is still safe and can be saved with no loss of data.
Network fault tolerance. If the network crashes while working on an Enterprise project the project is still safe. Transient disconnects won’t even be noticed, and lengthy disconnects still allow a local save of the project. Work can continue even without the network. When the network comes back up the project can be opened and automatic synchronization will occur.

Ability to use Internet. The high bandwidth requirements of older architectures mean that connecting over Internet to a remote DBMS is not usually realistic. With Manifold, the more efficient Enterprise architecture means that it is realistic in many cases to connect to an Enterprise server via corporate Internet links.

Off-line working capability. Older systems cannot work without a constant, "live" connection to the DBMS. Manifold allows transparent off-line work, such as with laptops when travelling away from a direct connection to an Enterprise server. Users can continue working while away from their corporate network.

Disadvantages of Shared Enterprise Server Storage

Although shared storage on Manifold Enterprise servers has many benefits, it does have some disadvantages as compared to either desktop models or classic server storage models:

- Exclusively Manifold usage. Enterprise servers cannot realistically be used by other GIS software. In exchange for the convenience of an easy to use environment the organization must use Manifold Enterprise Edition licenses on all clients. However, because Manifold Enterprise Edition also supports storage in "open" server storage models, users who desire interoperability with other GIS packages can also use linked drawings within classic server storage, such as within Oracle Spatial.
- No concurrent multi-user editing. Enterprise servers reduce the complexity of administration and reduce user skill set required by simplifying the shared usage of components. Many users can include shared components in their projects on a read-only basis, but only one user at a time can check out that shared component for editing. While checked out, that component may be edited only by that user. Once the component is checked back into the server, any changes made will propagate to all projects that use that component and a different user can then check out the component if desired.
- No support for area of interest windowing. When drawings are stored in Oracle databases Manifold can link only a portion of a drawing, the area of interest, into a project. It is usually the case when very large drawings are edited that each user is editing only a very small portion of the drawing. By linking in only that portion of interest, which is usually a few tens of megabytes of data, users can effectively work with drawings that are so large, potentially terabytes in size, that they could never be edited on a desktop machine. Because Enterprise Edition pulls the entire drawing from an Enterprise server it cannot be used to work with drawings that are larger than can fit within the performance characteristics of the desktop machine being used as a client.
- Greater complexity for programmatic access to objects. The Manifold API allows users to reach into Enterprise servers to obtain individual objects (that is, individual points, lines and areas) from within a drawing shared on the server, but doing so usually requires more skill than access to individual objects via storage methodologies that are more object oriented.
- Requirement for a database installation. This is really a disadvantage only as compared to the desktop storage model as the classic server storage model also requires a database. In earlier times the cost of a suitable enterprise-class DBMS might have been an obstacle, but given that DBMS vendors now provide free installations of the Express editions of the "Big 3" DBMS products, cost of a database installation is no longer a disadvantage. There is still the slight additional managerial overhead of installing a DBMS server on a machine.

Note that the above disadvantages can be avoided, if required, by using Manifold to store data in a server storage model using linked drawings. Manifold is perfectly happy supporting our work with whatever storage model we choose, be it local desktop storage, shared Manifold Enterprise server storage or linked server storage in the classic way. The advantages and disadvantages enumerated above apply to the shared Manifold Enterprise server model and are set forth so we can decide when we want to use that model.

In particular, none of the disadvantages apply when we use Manifold for server storage in the classic way, saving drawings and possibly images and surfaces within an enterprise-class DBMS like DB2, Oracle or SQL Server. We can even freely combine storage models within the same project to mix and match advantages and disadvantages as we see fit to optimize our storage strategy for a particular project or IMS application.

Enterprise Servers

Once a database has been created to which a Manifold component has been shared it is said to be an Enterprise server. Components stored within an Enterprise server are said to be shared components. Users accessing an Enterprise server using Enterprise Edition are said to be Enterprise users.
To create an Enterprise server we initialize the database and share at least component to it. Thereafter, any Enterprise user can fetch and use components from the Enterprise server. There are two ways for user to work with components saved within the Enterprise server:

- **Import** components from the Enterprise server. When importing components from the server a local copy of the component is made and work proceeds on that local copy without further connection to the server. Enterprise servers used in this way function simply as a centralized repository from which users may fetch components as desired for their local purposes.

- **Link** components from the Enterprise server. When a component is linked from the server the component continues to reside within the Enterprise server with controls on how that component can be used by different users at the same time. It is said to be a *shared* component. Enterprise Edition provides several ways of working with shared components that allow coordination between multiple simultaneous users.

Enterprise users can share components using the following commands, which appear in the context menu when right-clicking onto a component in the project pane:

- **Share** - Save a component into the Enterprise server. Use this command to load the Enterprise server with shared components.

- **Unshare** - Convert a shared component into a local component. The data from the Enterprise server will be downloaded to the local project and all connection to the Enterprise server for this component will be ended.

- **Check Out** - Gets the latest version of a shared component and makes it editable exclusively by user. Only one user at a time can check out a component. Other users can still link the component, but they will not see any changes we make until we check in that component.

- **Check In** - Saves the edited version of a component back to the Enterprise server and makes the component read-only in our project. When we are done editing a shared component we can **Check In** to save the changed version of the component to the Enterprise server. After we check in, any user getting that component will get the newly edited version.

- **Get Latest Version** - Fetches the latest version of a shared component. If we are working with a shared component and suspect that someone might have checked out the component we are using and altered it, we can use **Get Latest Version** to fetch the latest version.

- **Undo Check Out** - Enabled if we have used **Check Out** to get a component for editing. Abandons any changes we have made to a shared component on our local system, gets the latest version from the server and makes it read-only.

- **Link** - Appears in the File menu and in the Tools - Server Console dialog. Fetches the latest version of a shared component for use in our project. This opens a shared component in a read-only mode. We can view the component and allow it to participate in maps we create, but we cannot edit it.

- **Cached** - Shared components are cached by default. We can designate a component to be uncached by right clicking on a shared component in the project pane and choosing **Cached** from the pop-up context menu to toggle the check mark next to the **Cached** entry. Tool tips and the project pane status bar will show the cached/uncached status of each shared component. See the Cached and Uncached Components topic.

The above commands will be familiar to users of Microsoft Visual SourceSafe source code control system, if one understands **Link** to mean SourceSafe’s **Get**. The Manifold commands have, in fact, been deliberately designed to retain conceptual compatibility with similar commands used within source code control systems to resolve multi-user access and conflicts issues in complex development projects.

### Remote, Shared and Local Components

Components that are stored in an Enterprise server or which are linked in from other data sources are called **remote** components. Components in an Enterprise server that are linked into the current project are called **shared** components. Components that are stored within the project are called **local** components. A project can contain any combination of local or shared components. Shared components from many different Enterprise servers may be included within the same project.

The project pane shows shared components with a small **gray lock** icon next to the component icon to show they are read only. A component that is checked out by us will have a small **red check mark** icon. A component that is checked out by someone else will have a small **dark blue lock** icon. If someone else has checked out a component and edited it, the icon for that component will appear as a **dark red lock** icon. This indicates that if we want to work with the latest version of that component we should do a **Get Latest Version**.
Tool tips show the status of a shared component when the mouse hovers over it in the project pane. If a component is not checked out it will be reported as a shared component using the name of the Enterprise server data source. In the above example, the Roads Drawing component is shared on an Enterprise server data source called Enterprise example.

If we check out the component the icon turns to a red check mark and the tool tip reports it as being checked out.

If someone else checks out the component the icon turns to a black lock and the tool tip reports which user checked it out. In the above example, a user called Administrator on a machine called PROJECT has checked out the component. Tool tips report which user has checked a component so we can get in contact with them should we want them to check the component in so that we can check it out for editing.

If someone else checks out the component, changes it and then checks in the component, the icon turns to a dark red lock in our project. This lets us know it is time to get the latest version. The tool tip reports that the component is that it is outdated.

**Forms Cannot be Shared**

Forms components cannot be shared to an Enterprise server. The reason why not is that forms can contain programming controls which cannot be guaranteed to be available on the client computer. In some cases, users may employ licensed controls within their forms that are not redistributable to other computers without a license grant by the originating vendor.

**Nomenclature**

Although Enterprise components are brought in from the Enterprise server using a "link" command once they are in the project they are called "shared" components. Within Manifold, the term "linked" component is used to refer to tables that are linked into the project from an external data source other than an Enterprise server.

The main operational distinction between a linked table and a shared table is that a shared table is supervised by Enterprise Edition and it will be read-only unless it is checked out. An ordinary linked table will usually be read/write subject to the access controls of the host DBMS.
Project Capacity and Performance

When a component is shared to an Enterprise server, Manifold will upload a compressed version of the component into the Enterprise server. For large components, the process of compression can take a minute or more. When linking a component from an Enterprise server Manifold will fetch a local, read-only copy of the component into temporary local storage, decompressing it as it is fetched from the server. For large components, the decompression process can take a minute or more. Once a shared component has been linked into the project performance will be the same as if it were a local component.

Projects using shared components are subject to the same size limitation as other Manifold projects: for example, they cannot exceed a total capacity of 2 gigabytes. Enterprise servers can store as much information as can be handled by the host DBMS, possibly terabytes given the right hardware and software.

Compatible DBMS Products

Enterprise Edition has been developed using the latest editions of IBM DB2, Microsoft SQL Server (including SQL Server Express) and Oracle. These products are "supported" in that Enterprise Edition is designed to function using these three DBMS server products for Enterprise servers. Users purchasing Developer Tech Support incidents may ask questions about Enterprise Edition usage using those three DBMS servers as examples and may expect reasonably sensible answers.

Many other DBMS products may be used as Enterprise servers in addition to DB2, SQL Server and Oracle. For example, MySQL has been frequently used within manifold.net. Enterprise Edition even includes special code that works around a MySQL bug in a recent MySQL version that sometimes strips the last character in a stored section of binary data. There are many other fine DBMS products that will work well as Enterprise servers with Enterprise Edition. In general, any professional quality, multi-user, enterprise class DBMS that handles transactions well is a good candidate for use with Enterprise Edition. Some users even use Access 2000 (although this is discouraged since SQL Server Express is a profoundly better multi-user DBMS server).

MySQL users please note: Enterprise Edition requires that transactions support be enabled within MySQL when hosting an Enterprise server on MySQL. MySQL users should see the Manifold Knowledge Base articles on MySQL.

Enterprise Edition FAQ

Can I run Enterprise Edition without purchasing a DBMS system? Yes. You can install any of the three free DBMS packages and create Enterprise server databases within the DBMS installation thus created. You can also run Enterprise Edition without using Enterprise features (that is, like Professional) without any DBMS at all.

What is the best DBMS to use for an Enterprise Server? That depends upon the user's tastes and requirements. Each DBMS provided by Manifold is a superb, truly world-class DBMS, but all are different in various areas over which experienced DBMS administrators will passionately argue. Each DBMS has strengths in certain areas that would be a strong incentive for users interested in those strengths to choose that DBMS. See the Data Storage Strategies topic for a discussion. Users will often install and employ more than one of the three databases, although obviously there will be more to learn for the DBMS administrator if more than one system is used.

If we consider the limited question of just setting up a Manifold Enterprise server to use as a file cabinet without expectation of concurrent multi-user editing, then many users who are focused on capacity will install IBM DB2 Express-C Edition because it allows unlimited database size and will run using up to 4GB RAM and up to two processors. If maximum capacity in a free Express installation is the only consideration then clearly IBM DB2 Express-C has the edge.

But capacity as a Manifold Enterprise server using a free Express installation is not the only issue for many users. Many users will use mixed storage models where in addition to an Enterprise server they also want to use a classic geometry-storing centralized server to enable concurrent, multi-user editing of drawings or centralized storage of images. Or, they may have many existing applications that use Oracle or SQL Server. In that case, some other DBMS may be preferred. Any of the "big 3" DBMS packages may be used by Manifold as a spatial DBMS.

Users also may be considering the fit of the DBMS to their development environment. Organizations may realize their needs will far exceed the ability of even a very capable, free Express installation and may look to what DBMS they will eventually purchase in a full Enterprise configuration.
Even though Oracle Express does not include GeoRasters, other products from Oracle do, and once one expends the effort to learn how to be a DBMS administrator for a big-time DBMS there is a lot of wisdom to scaling up within the same vendor's product line. Users who are therefore just testing the waters with a free Express installation may well decide to go with Oracle even though the initial Express installation is limited in capacity to only 4 GB.

Finally, users who work with Microsoft programming in .NET or with Visual Studio will tend to favor SQL Server Express Edition despite the 4GB limitation on DBMS size because SQL Server Express is so tightly integrated with Windows development tools. Just as with Oracle, if users are using the Express edition to develop an application to be deployed on a much larger scale they will often choose SQL Server Express for development even though the ultimate application will require more than 4GB and will require a full license.

In all cases, when an organization's needs are so extensive that it is clear that the DBMS vendor's full DBMS product will be required it pays to have someone who really knows their DBMS technology examine the matter to make a decision which DBMS should be used. The three Express edition packages are a great starting point and a real testament to the power and competitiveness of these world-class DBMS vendors.

How is Enterprise Edition licensed? - See the License topic. Each computer on which Manifold System is installed must have a license for Manifold, except that for each license you may also install Manifold System on a notebook computer used by the primary licensee in addition to the main system used. You may install any of the three provided DBMS systems on any computer that will function as a server for Manifold. For example, you could have two Manifold licenses and thus operate Manifold on two different machines; however, you could also install SQL Server Express or DB2 Express-C or Oracle Express on several other machines to use as Enterprise servers. See the licenses within the installation packages for each of the three DBMS installations for details on allowed uses under the DBMS vendor's license.

Can I install IBM DB2, SQL Server or Oracle on a machine without installing Enterprise Edition? - Yes. You may install any of the DBMS packages provided on any machine you would like to configure as a server for Enterprise Edition or as a storage repository for tables for Professional Edition. You must use any such DBMS installation within the terms and conditions of the vendor's license that is packaged within the installation.

Does Enterprise Edition work with DBMS products other than the three supported? - Yes. It works with almost any DBMS that supports transactions and that has an ODBC driver. manifold.net has tested Enterprise Edition with Oracle, SQL Server and DB2 and will closely monitor these DBMS products as their vendors produce future editions. There are many other DBMS products with which Enterprise Edition will work; however, it is not possible for manifold.net to test them all or to keep track of new releases from all vendors.

Does Enterprise Edition read and write Oracle Spatial native formats? - Yes. See the Oracle Spatial Facilities topic. However, using Oracle spatial storage is a different thing than using an Enterprise server to store shared components. Using Enterprise Edition to set up Manifold Enterprise servers is a data storage model that shares the benefits of simpler, document-oriented storage models as well as the benefits of more complex, object-oriented server storage models.

Does Enterprise Edition read and write ArcSDE formats? - Yes, and ESRI Personal Geodatabase format as well. See the Spatial DBMS topic.

If more than five users attempt to work with SQL Server Express, will they lose data? - No. There is no longer a limitation to five users in SQL Server 2005 Express Edition as there used to be with the previous MSDE 2000 SQL Server Desktop Edition product. Even with the older MSDE 2000 product there would have been no loss of data, just much slower performance with more than five users.

Can I run Enterprise Edition with Windows 95? - No, and we sincerely hope that nobody ever tries. It is time to move on from early Windows editions and to install modern Windows editions like XP or Vista.

Can Enterprise Edition connect to an Enterprise server across Internet? - Yes. This is simply a matter of configuring the connection to the data source to work across Internet. Because Internet is normally slower than working in local area networks it is probably wise to retain the default cached storage option. There will then be some delay while fetching components initially from the server but thereafter work can proceed with local caching unless changes are made in shared components. That will greatly reduce the traffic passed back and forth over Internet.

Troubleshooting
To be sure your Enterprise Edition installation is working correctly, create an Enterprise server using SQL Server Express on the same machine, using an Administrator login for all work. The usual cause of any problems sharing components or otherwise using Enterprise servers are errors in the configuration or operation of the database server, or errors in the configuration or use of access permissions. It's the same familiar story well known to network and database administrators: one can configure a beautiful installation, but if a given user does not have access permissions to use it problems will occur. Ultimately, one cannot administer an Enterprise Edition administration if one does not know how to administer the host Windows operating system as well as the DBMS being used to host the Enterprise server.

See Also

Spatial DBMS
Data Storage Strategies
Creating an Enterprise Server
Working with Enterprise Edition
Cached and Uncached Components
IBM DB2 Express-C Edition
Oracle Express Edition
SQL Server Express Edition
Administering Enterprise Servers
Server Console
Creating an Enterprise Server

Create an Enterprise server by creating a database in the database provider and then sharing a component into that database. This topic uses Microsoft SQL Server 2005 Express Edition as an example. Procedures for other database providers will be similar.

Creating an Enterprise server is easy:

- Create a new SQL Server database and set up the necessary permissions as described in SQL Server Express Edition.
- Create an ODBC data source to connect to that database. This can be done either within Manifold or using the Windows ODBC Data Sources applet in the Control Panel. (There is a shortcut to eliminate this step mentioned at the end of this topic.)
- Within Manifold, share at least one component into the new data source. Sharing the first component into the data source sets up the Enterprise server.

The lengthiest part of the process is creating a new ODBC data source. This involves stepping through several dialog pages. In the following example we create a new Enterprise server using SQL Server Enterprise Manager and Manifold.

Creating an Enterprise server database:

1. (Assuming an instance of SQL Server Express named SQLEXPRESS installed on a machine named MYSERVER.) Create a new SQL Server database named example and set up the necessary permissions as described in SQL Server Express Edition.
2. Log in to a machine that can access the database, using a user account with read and write permissions in the database.
3. Launch Manifold.
4. Create a blank drawing. Right click the drawing in the Project pane and select Share.
5. In the Share dialog, press the ... browse button to view available data sources. We will assume that no data sources have yet been created for the desired instance of SQL Server and will create a new data source.
6. In the Select Data Source dialog, switch to the Machine Data Source tab and click New.
7. In the Create New Data Source dialog, select User Data Source, click Next, then in the list of drivers select SQL Server, click Next, and finally click Finish.
8. In the Create a New Data Source to SQL Server dialog, set Name to something you can remember, for instance, Example Enterprise server, set Server to refer to MYSERVER\SQLEXPRESS, and click Next. Confirm that you will be connecting With Windows NT authentication using the network login ID and click Next. Check Change the default database to option and set the database to example, then click Next. Click Finish.
9. In the ODBC Microsoft SQL Server Setup dialog, click Test Data Source to make sure the data source works, then click OK twice.
10. In the Select Data Source dialog, select the newly created data source and click OK.
11. In the SQL Server Login dialog, confirm the choice of connection options and click OK.

(back in the Share dialog)

12. In the Share dialog, press OK. This will connect to the database, create the infrastructure tables for keeping track of shared components, and upload the drawing. The Project pane will show a "lock" icon near the drawing to indicate that it is now shared on an Enterprise server. If we save the project at this time, the component will be a shared component. Depending on whether we have caching turned on or off the cache will be saved locally. See theCached and Uncached Components topic for information on caching.

We have just created an Enterprise server that has one component in it. Note that everything in steps 6 through 11 was the routine Windows task of setting up a new ODBC data source. Creating a new SQL Server database was almost easier than creating a data source. Sharing a component (the actual Manifold part of the exercise) was very easy as well.
By definition, a new Enterprise server has at least one component in it. Once the Enterprise server is created we can Share (upload) more components into it and we can use those components in new projects.

Using an Enterprise server

1. Launch Manifold and open a project.
2. Choose Tools - Server Console to launch the Server Console dialog that allows browsing Enterprise servers.
3. In the Server box, the data source connection string that was last used will appear. Specify a new connection string manually or by clicking the [...] browse button. Press the Refresh button at any time to reconnect to the server and re-read its contents.
4. Click on the component desired and the toolbar buttons to Import or Link will be enabled.
5. Press Import to import the component into the project. This fetches a copy of the component from the Enterprise server into local project storage. Press Link to link the component into the project, leaving it stored within the Enterprise server and under control of the Enterprise server.

Components that have been linked from the Enterprise server into the current project will appear with a chain icon in the Server Console. They will also appear in the project pane with a lock icon when shared (that is, read only). Components in the project that are not provided by an Enterprise server will not have an icon. When the mouse moves over a component in the project pane a tool tip will read out the status of the component, such as the name of the user who has checked out the component.

If a component is checked out it can be deleted via the Server Console only by the user who has checked it out. If a component is not checked out it may be deleted by anyone via the Server Console. Components may be deleted, but they may not be renamed.

Folders

The New Folder button in the Server Console toolbar creates folders within the Enterprise server. Use folders to organize components within the Enterprise server and to keep the Enterprise server manageable.

This is especially important when Enterprise servers are very large and contain many components. Folders may be created anywhere within the tree hierarchy, and folders can be created within other folders. For example, we might have folders for countries, folders within those folders for provinces and then folders within those for different classes of information such as hydrography, transportation, boundaries and so on.

To move files between folders, drag components and drop them into a folder (drop them onto the folder icon itself and not the hierarchy beneath the folder). Folders may be created within other folders. If a folder is highlighted in the Enterprise server pane and the New Folder command is issued, the new folder will be created within the highlighted folder. Folders may be dragged and dropped into other folders. To move a folder out from within another folder, drag it and drop it into the "white space" of the Enterprise server outside another folder. Folders may not be renamed, but they may be deleted. To delete a folder that contains components the components involved must be checked out.

There is no limit on the number of folders within an Enterprise server, but the number of components within a single folder is limited to approximately 1000 components. As a practical matter, if one is working with such a large number of components it is wise to organize them within subfolders.

Simplified Syntax for SQL Server Connections

Enterprise dialogs allow a simplified connection string syntax for SQL Server databases as an option to full ODBC syntax. \\
connects to the specified database on the given system. \systemname:database\ connects to the default database on the given system. The connection established with a simplified connection string that uses Windows integrated security (suitable for users working in Windows Server 2003, Windows XP and Windows 2000). Using simplified connection strings is a fast way to connect to SQL Server or to SQL Server Express using default options without dealing with the ODBC dialogs.

For example, if we have installed SQL Server on a system called PROJECTS and we have created a database called hydrography to use as our Enterprise server database, we could connect in the Server Console by entering \projects\hydrography into the Server box of the Server Console and then pressing the Refresh button.
SQL Server Express

Microsoft provides free download of a complete installation package for Microsoft's SQL Server Express, the same engine that powers SQL Server. Installing SQL Server Express allows high performance, centralized DBMS storage via Enterprise servers using SQL Server capabilities without any additional purchase required. See the SQL Server Express Edition topic for more.

Connection String Security

When Manifold saves the last used connection string in between Manifold sessions, the connection string is stored in encrypted form. This security measure prevents intruders from learning your connection strings by attacking the Windows registry or other commonly available resources.

Some connection strings to connect to Enterprise servers may use passwords in PWD entries within the connection string. For additional security, Enterprise dialogs conceal passwords that occur within connection strings by replacing the password’s characters with * asterisks. See your ODBC documentation to learn about connection strings and PWD entries in connection strings.

When a user enters a connection string containing a password and presses the Refresh button Manifold analyzes the connection string to identify the password, saves the password in encrypted form in internal memory and then substitutes the password with asterisks in the edit box so that the password does not appear in plain text.

If the user later modifies the connection string (say, by changing the name of the server) and presses Refresh, Manifold knows to use the internal, saved password in place of the asterisks when connecting to the data source. If the user modifies the connection string so that it no longer contains a PWD entry, no password is used. If the user modifies the content of the PWD entry to some value that does not entirely consist of asterisks, the system will accept the new value as the new password. The above works with the Browse button as well.

Notes

The simplified syntax available for SQL Server connections means that we really did not need to create an ODBC connection to work with an Enterprise server hosted on SQL Server as set forth in the beginning of this topic. After creating a database in SQL Server using the SQL Server Enterprise Manager and granting users the desired permissions, we could have shared the component by simply entering \systemname:example into the Share on box when we shared the first component into the Enterprise server. After that, we could enter \systemname:example into the Server box of the Server Console to work with that Enterprise server.

Using simplified syntax is somewhat easier than configuring an ODBC source as described in this topic, but it only works with SQL Server. This topic set forth the full procedure to illustrate how ODBC sources can be configured from the browse button in the Share dialog, to provide an example that may be more useful when using databases other than SQL Server which require configuration of an ODBC source.

The short examples given above show one database created within SQL Server. We could create many databases within each SQL Server installation and use them all simultaneously. Each database is used as a separate Enterprise server.

For example, suppose on the machine called PROJECTS we install SQL Server and then create databases called hydrography, boundaries, transportation and utilities. We could work with each of these in the Server Console as four different Enterprise servers using the shorthand connection syntax of \projects:hydrography, \projects:boundaries, \projects:transportation and \projects:utilities. We could include shared components from each of these in the same project.

See Also

SQL Server Express Edition
Working with Enterprise Edition
Cached and Uncached Components
Working with Enterprise Edition

Setting aside the creation of Enterprise server databases and the ODBC data sources used to connect to them, working with Manifold System Enterprise Edition is very similar to Manifold System Professional Edition.

There are two key dialogs used for interactions with Enterprise servers:

- The Tools - Server Console dialog shows a catalog of components available within a given Enterprise server. Use the Server Console to import or link components from an Enterprise server. The Server Console is also used to organize Enterprise servers by creating folders within the server and arranging components within those folders. Use the Server Console to delete components from the Enterprise server. Deleting a component will also remove all maps containing it as a layer.

- The project pane shows components in the current project just as it does in Manifold System Professional Edition. Right click onto a shared component to Get Latest Version, Check Out, Undo Check Out or Check In an existing remote component or to use Share to upload a component to the Enterprise server.

Adding Components to an Enterprise Server

To add components to an Enterprise server, right click on a component in the project pane and choose Share. The process is called sharing a component to the server. The component will be uploaded to the Enterprise server and will appear in the project as a shared, read-only component. Sharing a component will share all dependent components as well. For example, if we share a map to an Enterprise server all of the components that are layers in the map will also be shared to that server.

When sharing a component to the server the component is uploaded to the Enterprise server in compressed form, growing the storage requirements of the Enterprise server by the compressed size of the component. Compression achieved by Enterprise servers will vary depending on the component with typical compression factors achieved being those similar to compression achieved by "zip" utilities, the same as if the components were saved in a compressed .map project file.

Sharing a subsidiary component to an Enterprise server will also share the parent component to the Enterprise server. For example, sharing a terrain or a profile will also share the terrain's or the profile's parent surface as well. Sharing an elevation will also share the elevation's profile and the profile's surface as well.

Removing Components from an Enterprise Server

Use the Server Console to remove components from an Enterprise server. Click on a component to highlight it and then press the Remove button. We cannot remove a component that is checked out. If a component is linked into a project removing it from the server will also remove it from the project.

Removing a component from the server will remove all components in our project that are bound to that component. For example, removing a map from the server will remove the map, but will leave components that were constituent layers of the map (the map depends on the layers but not vice versa). On the other hand, removing a component will remove any map from the server that contains that component as a layer. Removing a drawing will also remove the drawing's table and vice versa because a drawing and the drawing's table are inextricably bound to each other.

Getting Shared Components

Components may be imported from any of the file formats supported by Manifold. They may also be imported or linked from an Enterprise server by using either the File - Import or File - Link commands or via the Tools - Server Console dialog.

When a remote component is linked into the current project it appears in the project pane with a "lock" icon to show it is a shared component that is read-only. If it is opened, it may be viewed or participate in a map but it cannot be modified in any way. If we open it in a window the format toolbar, the selection toolbar and other editing toolbars will not be enabled.

Modifying a Shared Component
To modify a shared component we right click on it and choose Check Out. When a component is checked out it may be modified. All toolbars will be enabled for that component. As long as we have a component checked out from an Enterprise server, no other user can check out that component. Another user can get it by linking it into their project, but they cannot check it out to make modifications until we check it back into the server. When done modifying a component we should right click on it and Check In. This will upload the changes to the server and make the component a read-only shared component.

If we would like to abandon any modifications made, we right click on a component and choose Undo Check Out. This abandons any changes and gets the latest version of the component from the server. Checked out components cannot be deleted from a project pane. To delete a checked out component, either check it in or undo the check out and then delete it.

Manifold tries to provide the greatest range of modifications possible in a component without checking out dependent components. This adds flexibility at the cost of requiring some additional thought. For example, if we check out a drawing we will be able to format that drawing but we will not be able to add new objects (points, lines, areas) unless we also check out the drawing’s table.

Note that if another user has already checked out a component that we wish to modify we will not be able to Check Out the component. We will have to wait until that other user has checked it back in.

When exiting Manifold or closing a project, the system will remind us if we have any shared components that are still checked out. It is possible to close Manifold while a component is still checked out. We may want to keep a component checked out until the next time we are able to work with Manifold because we have not yet finished modifying it.

**Getting Changes**

If someone else checks out a shared component we will not see any modifications they make until they check it in and we choose Get Latest Version. This will update the component in our project with the latest version from the Enterprise server.

**Working with Maps**

Maps may be shared to an Enterprise server subject to the following conditions:

- A shared map can only contain layers from the same Enterprise server on which it is shared.
- Adding an existing layer to a shared map requires that the existing layer a) is shared on the same Enterprise server with the map and b) is checked out.

Important: components not fulfilling the above criteria are not shown in the Add Existing Layer dialog and will refuse to drop into a map when dragged from the Project pane.

- Removing existing layers from a shared map does not require anything except that the map itself must be checked out,
- Adding new layers to a shared map is not allowed.
- Local maps can contain both local components and shared components (possibly from different servers).
- Removing a shared drawing, image, labels or surface component from a project will remove all shared maps containing it as a layer.

**Sharing Linked Images**

Sharing a linked image to an Enterprise server places the linking information into the Enterprise server and not the actual image data. Importing or linking such a shared linked image to a local project will re-establish the connection to the relevant file or server. See the Linked Images topic.

**Miscellaneous Dependencies**
Because shared components are usually read-only, numerous dependencies arise between components that are related to other components. Non-trivial dependencies that will become apparent as a result of automatic disabling of menus and toolbar buttons include:

- Adding or removing objects in a shared drawing requires checking out both the drawing and the table bound to the drawing (which is also shared). Since most drawing transforms either create or remove objects, they also require checking out both the drawing and the drawing's table. Editing objects in a shared drawing using visual editing tools (so long as objects are modified but not added or removed) requires checking out the drawing, but not the table.
- The Transfer Selection dialog does not allow transferring a selection to shared components that have not been checked out.
- Removing records from a table bound to a shared drawing (hence also shared) is similar to removing objects from that drawing and requires checking out both the table and the drawing it is bound to.
- Removing a table column used in drawing formatting (assuming both the drawing and the table are shared) requires checking out both table and drawing.
- Adding a relation to a shared table can only go to a table shared on the same Enterprise server and requires checking out both tables. The Add Relation dialog launched on a shared table hides local tables and tables shared on other Enterprise servers. The Add Relation dialog launched on a local table hides all shared tables.
- Shared components cannot be renamed.
- To preserve naming consistency, remote components cannot be linked into a project if they have the same name as a local component already in the project. Likewise, a local component cannot be shared to an Enterprise server if the server already has a remote component of the same name. Rename components as necessary to make room for like-named components that are to be linked or shared.
- A shared map does not allow dragging layer tabs to new positions unless it is checked out.
- Unsharing a component that "owns" other component unshares all owned components. For example, unsharing a drawing also unshares its table, any scripts bound to the table and any labels bound to the drawing. Similarly, unsharing a component that "is referred to" by other components unshares all referring components. For example, unsharing a drawing also unshares any maps containing the drawing as a layer, any tables linked to the drawing's table with relations, any components owned by such tables (such as scripts) and so on.

Relations and Shared Tables

For advanced users, it may be useful to consider how relations between shared tables can and cannot be formed.

Manifold can establish a relation between an ordinary table and a linked table, between two linked tables, between two shared tables located on the same Enterprise server, but not between an ordinary table and a shared table, or between two shared tables located on different Enterprise servers.

Let's do a thought experiment to consider what could happen if a relation between an ordinary table and a shared table were allowed. Suppose we check out a shared table, create a relation between this shared table and an ordinary table, bring some columns from the ordinary table into the shared table, and then check in the shared table. We then go to another machine and bring the shared table into a new project. On the new machine, Manifold will not be able to display the data in the columns brought in from the ordinary table we used on the first machine, since that table is not in the project we created on the second machine.

Even if the project on the second machine happens to contain a table with the same name as the ordinary table on the first machine, there is no guarantee that the shared table will display the same data on the first and second machines. To guard against situations like this we cannot form a relation between an ordinary table and a shared table.

If it was possible to create a relation between two shared tables located on different Enterprise servers we could encounter similar problems. If this were allowed there would be nothing stopping us from linking to the first table, but not to the second table, and creating a situation where the first table would not have any way to display data from the second table. Note that this is different from the case where both tables are on the same Enterprise server, since in that case the system will automatically link in the second table if we attempt to link to the first table.

Saving Projects - Cached Components

Enterprise projects are saved in the usual way. Components may be saved as cached or as uncached components. When the project is opened again the links to the Enterprise server will be automatically re-established if the project is opened on the same system, where the same connection string to the ODBC data
source representing the Enterprise server can work. See the Cached and Uncached Components topic for additional information.

When saving a project that contains a cached checked out component, that component will remain checked out until that project is opened again and the component checked in. To see how this works, check out a component and then save the project and close Manifold. Launch Manifold again and link the component from the Enterprise server into the new project. It will be reported as checked out by some other user because it is still checked out in the previous project that was saved.

This is good, because it allows people to take breaks while reserving the component and their current work state by leaving it checked out. In fact, one can check out a component and save the project and then open a second project and link the component from the server into that second project. In the saved project the component is checked out. In the new, open project the component is shown as checked out by another user.

This, too, is sensible because we may be working with the same component in several different projects at the same time. We might be in the middle of editing it in one project yet have to set aside our work temporarily to work on a different project that also uses that component. When we set aside our editing session we can save the project with the component in a checked out state. When we open another project that also uses that component it will appear in the project as checked out by another user. The component will be seen in the project pane with a dark lock icon, the usual icon showing it is checked out by a different user (the tool tip info will show it is checked out by us).

When we open the Server Console it will show components that are checked out by us in different projects with a red link icon. This is to remind us that we have some components checked out in projects other than the one on which we are currently working.

It could well be that we check out a component, save the project, set aside our work and then later forget we have saved the project with a checked out component. It might happen that when we see the red link icon for a component in the Server Console we won't remember in which project .map file it has been saved in a checked out state, or even whether or not we have deleted the .map file in which it was saved in a checked out state. At such times we can use the Undo Check Out button in the Server Console to undo the check out status of that component. This command will reset the state of the component on the Enterprise server to checked in. The next time the project containing the component in a checked out state is opened or refreshed, Manifold will automatically execute an Undo Check Out on that component in the project state and will cause any changes made since the last check in to be discarded.

Users with experience participating in large teams will predict that some users may check out a component and then depart for vacation or otherwise instill a desire in Administrators to forcibly rollback a checkout for a component that the server shows is checked out in someone’s project. Undoing check outs made by users other than the current user is unsafe and should be done only by the DB administrator. See the Administering Enterprise Servers topic for notes on forced rollback (undo check out) of components checked out by someone else.

Note that we cannot Close a project if it contains checked out components that have been checked out since the last save.

Saving Projects - Uncached Components

Saving projects that contain uncached components is the same as saving projects with cached components, except that in the case of uncached components we cannot save a project that contains checked out components. The reason is simple: since an uncached component is always reloaded from the Enterprise server when a project is opened, it makes no sense to save a partially edited version that will be replaced the moment the project is reopened. Likewise, one cannot Close a project that contains a checked out, uncached component.

Capacity

Enterprise projects are limited to a maximum of 2 gigabytes of components in uncompressed form. There is no limit to the number of components that may be saved on the Enterprise server except for whatever capacity limitations are imposed by the database provider used to host the Enterprise server.

Security
User access permissions in Enterprise Edition are set by the user access permissions applicable to the database that hosts the Enterprise server. Most database management systems, such as SQL Server, have numerous options for database security in combination with the operating system security model as well as various hierarchical rules controlling propagation of permissions between different database objects. It is therefore essential to understand thoroughly the Windows security model as well as the security model used by the database system.

**Project Pane Updates**

By default, Manifold will contact the Enterprise server every 120 seconds to see if any shared components have been checked out or checked back in by another user, updating the status icons in the project pane accordingly. This automatic refresh may be disabled in **Tools - Options - Server Storage** by unchecking the **Refresh state of shared components** box. The refresh time may also be set to a value different than 120 seconds.

**Tech Tip**

We would often like to save comments with a shared component on the Enterprise server so that whenever someone gets the component they will automatically get our comments with it. This can be done by saving any notes desired into the **Description** box of a component's **Properties** dialog.

**Notes**

Why must an existing layer be checked out if it is to be added to a shared map? The existing layer must be checked out to protect against the following scenario:

- An Enterprise server contains map M and drawings D1, D2 and D3. Drawings D1 and D2 are layers of M and all components are checked-in.
- User U1 check outs M and adds D3 to the map,
- User U2 deletes D3 from the server,
- User U1 checks in M, which is now invalid.
Cached and Uncached Components

This topic applies to Enterprise Edition only. If you have not installed Enterprise Edition you will not be able to use the features discussed.

When working with Enterprise Edition we will normally want to save our work in a project by saving the project to a .map file in the usual way. Enterprise Edition has two modes in which shared components can be saved when a project is saved to a .map file. Shared components can be cached or uncached for the purpose of saving projects.

- **Cached components** - When a component is marked Cached the Enterprise cache for that component will be written into the .map project file when the project is saved. This will grow the .map file by the size of the cached component.

- **Uncached components** - When a component is not marked Cached the Enterprise cache for that component will not be written into the .map file. Instead, only a link to the shared component on the Enterprise server for that component will be saved.

Shared components are cached by default. We can designate a component to be uncached by right clicking on a shared component in the project pane and choosingCached from the pop-up context menu to toggle the check mark next to the Cached entry. Tool tips and the project pane status bar will show the cached/uncached status of each shared component.

The word "cache" is a computer term that refers to use of local memory that may be rapidly accessed. When Manifold first links a shared component into a project from an Enterprise server it fetches the component from the server into an Enterprise cache maintained in local memory, both in RAM and in temp files on local disk. The Enterprise cache always contains a snapshot of all components with which we are working whether they are shared or local. The Enterprise cache allows operations in local memory whenever possible to greatly reduce transactions with the Enterprise server. This increases speed because it greatly reduces the interactions between the client and the server, so our desktop (the client) is not slowed down by bottlenecks at the server.

When we save a project, Manifold checks the Cached status of each shared component in the project. If the Cached setting is checked, the Enterprise cache for that component will be saved into the .map project file that is created when the project is saved. As a practical matter, saving the Enterprise cache into the project for a component is equivalent to saving a controlled copy of the component into the project’s .map file. Doing so will grow the .map file by the size of the component.

If the shared component is uncached, that is, the Cached setting is not checked, then the project’s .map file will contain only a link back to the remote component on the Enterprise server. Saving a project that contains uncached, shared components saves no data from shared component within the project’s .map file, so it will not grow the size of the .map file by the size of the component.

In both cases, regardless of whether a shared component is cached or uncached, the Enterprise status of the shared component will be preserved when the project is next opened. For example, suppose we create a new project, link into it several shared components, check out one of those components and save the project. We could close Manifold, go away for lunch, come back later and open the project and the component will still be checked out. This will be true whether or not the components were cached or uncached.

Advantages and Disadvantages

Some readers may wonder: If checking the Cached setting for a shared component causes a copy of it to be saved within the .map file, doesn’t that defeat the purpose of an Enterprise server? That is only so if one imagines the purpose of an Enterprise server to be limited to saving disk space. In an era of essentially free disk storage capacity (due to the very large sizes of modern disk drives and their very low costs), the greatest benefit of centralized storage on an Enterprise server is not saving disk space, it is making work more efficient by allowing collaborative efforts within teams and by providing centralized archives and version control.

Caching components and saving them to project files does not interfere in any way with the component discipline implied by Enterprise Edition’s sharing, check out and check in procedures. The client side of Enterprise Edition will enforce all dependencies just as if components were not cached. Caching allows faster client startup, more efficient server operation and greater operational flexibility and safety.

Caching allows greater performance when a project is opened because the constituents of that project need not be fetched from the Enterprise server. When a project consists of large components the time required for decompressing and fetching a component from the Enterprise server could be a long time even if no one else is
trying to fetch components from the server at the same time. Caching components reduces the need to interact with the Enterprise server to only those times when components are changed.

Advantages of Cached Components

- Reduces load on the Enterprise server, allowing the same system to serve more users.
- Faster opening of projects when the Enterprise server is loaded by other user requests at the same time, such as at the beginning of a business day when many users at once may launch projects that fetch data from an Enterprise server.
- Work can continue on projects when the Enterprise server is inaccessible. For example, one could begin a project on a notebook computer connected to a local area network, save the project and then continue working with the project while travelling on an airplane.
- Reduced latency allows working with larger projects if the Enterprise server is accessed over an Intranet or the Internet.
- Automatic redundancy safeguards important data. If the Enterprise server crashes we will not lose any data if a project has been saved with cached components.
- Professional Edition can transparently handle cached shared components, so it can open a saved Enterprise project if it includes only local and cached shared components. This allows easy data and project interchange between a work group that includes both Enterprise and Professional users.
- Data exchange becomes simpler, because one can send a .map file that contains cached components to another user. Even if the Enterprise server cannot be contacted at least the components will be available in their last-saved state.
- Projects using cached components can be saved with cached components in a checked out state. This allows checking out a component, performing some edits and then setting the work aside in a saved project to be finished at a later time.

Disadvantages of Cached Components

- Cached components require more disk space when projects are saved, because each saved project contains the Enterprise cache image of the cached shared component.
- When a project is opened users must get the latest version of components that have been changed by other users since the project was last opened.

Advantages of Uncached Components

- Opening a project automatically gets the latest version of all uncached components. Because uncached components are fetched in their entirety every time a project is opened they are automatically up to date.
- If many common files are in use throughout an organization, not caching components will assure that only one copy's worth of disk space is used for a shared component throughout an organization: the space required on the Enterprise server for that component.

Disadvantages of Uncached Components

- Uncached components exist in only one location: the Enterprise server. If a connection to the server cannot be established or if the server crashes the components will be deleted from the project. It is therefore essential to establish and maintain a backup regime to backup the database on a regular basis. However, setting up automatic backup of databases is often a chore neglected by the experienced administrator and beyond the skill set of a novice.
- Delays caused during "rush hour" in the morning or at the end of a work day can lead to user complaints, since few organizations scale their hardware to provide acceptable performance during peak loads. Because disk space is a lot cheaper than system throughput it is far cheaper and easier to deploy slightly larger hard disks for user desktops than it is to scale a central server to handle huge loads that appear for only a fraction of the work day.
- Professional Edition users cannot use projects containing uncached components.
- The need to fetch uncached components every time the project is open will limit the size of projects that are practical to share via Internet.
- Data exchange becomes more complicated because it is generally not possible to simply exchange .map files, since the connection strings to the Enterprise server will not likely be the same for a different user. If we send a .map file containing shared components to someone else they will not be able to see or use the shared components unless they are able to use exactly the same connection string to connect to the data source for that Enterprise server. This is highly unlikely unless both users are working on the same machine.
Projects containing uncached components cannot be saved if any of the uncached components is checked out. This means that all desired edits must be accomplished and the component checked in before the project is saved.

From the points above it can be seen that the main benefit of using uncached components is a reduction in disk space used when the same component appears in many projects. For a single user working with many projects this can be a worthwhile savings, since a single user will be working with only one project at a time.

However, if the objective is to reduce disk space because many users have included the same shared component in their projects the tradeoff is not so simple. In this case, the reduction in disk storage comes at the cost of a dramatic increase in transactions with the Enterprise server when many users open their projects and thus a greatly increased peak-processing requirement during busy times of the day.

If components are uncached, one must invest in a reliable, fast machine to host the Enterprise server. It must be reliable because if it fails our data disappears. It must be fast because multiple user demands will be more likely when every time a project is opened every uncached component must be loaded from the server.

In contrast, if we are willing to take advantage of large, cheap hard disks and use cached components we can use a relatively inexpensive machine for our Enterprise server that need not be particularly fast. User accesses to the Enterprise server will rarely happen simultaneously because components will be read or written only when they are changed. The great majority of projects will load shared components only when they are first created. Thereafter, user load will be negligible since the server is consulted only in brief contacts to check the status of components. On occasion when a component is changed there will be some activity as a new version of a component is checked in and other users get the latest version.

Of course, one can use a mix of cached and uncached components in a project to balance the advantages and disadvantages of caching in a particular project.

Default Settings

In Tools - Options - Server Storage the Cache newly shared components option specifies whether newly shared components will be cached or uncached. By default, this option is checked so that newly shared components are cached. If this option is not checked, newly shared components will be uncached.

Notes

When working with uncached components Manifold will load the uncached components when a project is opened; however, it also most load the uncached component from the Enterprise server once more after saving a project. Both operations are internal and are transparent to the user except for the time required to fetch an uncached component from the Enterprise server. Users will normally expect a delay while components are fetched from the Enterprise server when a project is opened; however, a similar delay when saving a project may come as a surprise. However, the re-load of cached components upon a project save is required to maintain cache synchronization.

See Also

Working with Enterprise Edition
Administering Enterprise Servers
Administering Enterprise Servers

Enterprise servers used with Enterprise Edition are based on the host database, so routine administration of Enterprise servers mainly consists of the usual tasks involved in administering the host database. For information on administering your host database (backups and similar tasks) see the documentation provided by your database vendor. This topic discusses issues specific to the interplay of Enterprise servers and saved projects.

Checked Out Components

The main administrative issue arising when working with Enterprise servers occurs when one user has checked out a component that another user wishes to edit. If one user has checked out a component for editing no other user can edit the component until it is checked back in. If both users are online at the same time the situation is simple to resolve - the user who wants the component emails, instant messages or calls the user who has it checked out and asks that the component be checked back in.

Saved Projects and Cached Components

A more complex scenario emerges if a user checks out a cached component, saves the project and then the component needs to be used for editing. This situation usually occurs in three forms.

- The same user has checked the component out and saved it in one project file, but needs to edit it in another project file. In this case, the user can open the Server Console and use Undo Check Out to undo the check out. This will show the component as checked in on the server and will abandon any edits to the component that may have been made in the project where it was checked out. This works even in cases where the user cannot remember in which project file the checked out component can be saved.

- One user has checked out a component and saved it in a project file but another user needs to check out the component. In this case, if the user with the checked out component can be located other users must negotiate with him or her to get the needed component checked back in.

- At times users will check out cached components, save them in a project and then become unavailable, perhaps going off on vacation or departing the organization. In such cases a database administrator will have to do a forced rollback of the component check out.

Undo Check Out compared to Check In

The Undo Check Out command is not the same as a Check In command. When a component is checked out if it is checked in any edits made while it was checked out will be saved to the component. When an Undo Check Out is done, any edits made while it was checked out will be abandoned.

Forced Rollbacks

Undoing a check out is also called a rollback. Users can undo a check out for components they themselves have checked out by using the Server Console’s Undo Check Out command. This command will mark the component as checked in on the server so that the project can update itself the next time it is refreshed. Projects are refreshed whenever:

- The project is opened.
- The user presses the Refresh button on the project pane toolbar.
- The refresh timeout interval specified in Tools - Options passes.

A more complex situation arises when an administrator must undo a check out for a component when a user who checked out that component is not available to use Undo Check Out in the Server Console. In that case, a manual Undo Check Out must be performed within the Enterprise server’s database structure. This is done by manually opening the Enterprise server database and editing fields within the database tables using whatever tool is normally used to make edits to the database.

To force a rollback of a component in an Enterprise server:
1. Using a suitable tool, connect to the database containing the Enterprise server.
2. Open the mfd_root table.
3. In the mfd_name field, find the names of the components for which check out is to be undone. Each component will be one record.
4. For each such record, change the values of the mfd_user and mfd_user_info fields to either blank strings or NULLs.
5. Commit changes.

Very important: Make sure not to edit the system record by accident. This record has "system" in the mfd_type field.

Note that the above procedure requires the ability and skill set to be able to open database tables and to edit them. Most DBMS products used as Enterprise servers will have such interactive tools. If a system such as SQL Server Express is used that does not provide interactive tools, one can always link the mfd_root table into a Manifold project using File - Link - Table (OLE DB) and then browse the table and make changes to the mfd_user and mfd_user_info fields in the linked table. Note that OLE DB drivers for most databases do not show a dynamic view of tables: they only show the tables at the moment the connection was made. To see the results of edits, therefore, one must either do a View - Refresh or close the connection and reconnect.

Orphaned Check Outs

An orphaned check out occurs if a component has been checked out on an Enterprise server without being checked out in a project. This happens in two cases:

- A user checks out a cached component and saves it in a project. Later on the user deletes the project file, either by accident or deliberately (forgetting it contains a saved, checked out component).
- A user checks out a component while working on a project when a crash occurs before the project can be saved or the component checked in. This might happen during an electrical power failure or other unexpected crash.

Orphaned check outs are easily resolved by the user launching Manifold, opening the Server Console and issuing an Undo Check Out command for the component. If the user who has an orphaned check out is no longer available, the database administrator can follow the procedure noted above for a forced rollback.

Cheap Insurance

As with any centralized DBMS, if the Enterprise server database is damaged the data it contains can be lost. It is very important to use a robust DBMS within a robust operating system that is hosted on reliable hardware. The following tips will increase reliability of Enterprise servers:

- Choose a reliable DBMS. Use a real DBMS with reliable transaction capabilities. For example, although Access 2000 is said to work with Enterprise Edition it is a poor choice because of its relatively lower reliability in multi-user environments. Use SQL Server Express instead. It is a free download from Microsoft. SQL Server Express has the full reliability of SQL Server in multi-user environments.
- Use Windows 7 or Server 2008. Both of these systems are so much more reliable than earlier Windows editions that they should be your only choices for server applications.
- Choose reliable hardware. Insist on a high quality motherboard for the server and run it conservatively (avoid overclocking processors and other "hot rod" alterations). Consider choosing a motherboard with built-in hardware RAID so that two identical disk drives can be installed for a hardware mirror. If one disk drive fails, the other will enable continued operation until a second drive can be installed.
- Install service packs and other updates when they are issued by the DBMS vendor and Microsoft. Install the latest service packs for Manifold as well.
- Install lots of RAM memory. Working with many users in complex DBMS products with small amounts of RAM is a great way to find previously undiscovered bugs in DBMS products.
- Configure the DBMS for automatic backup to another computer system on the local area network. Systems like SQL Server can be configured to automatically save a backup to a different location.
- Install an uninterruptable power supply (UPS) to power your server through unexpected electrical power outages and to shut down the server down if the power outage continues past the reserve power of the UPS.

Troubleshooting
To be sure your Enterprise Edition installation is working correctly, create an Enterprise server using SQL Server Express on the same machine, using an Administrator login for all work. The usual cause of any problems sharing components or otherwise using Enterprise servers are errors in the configuration or operation of the database server, or errors in the configuration or use of access permissions. It's the same familiar story well known to network and database administrators: one can configure a beautiful installation, but if a given user does not have access permissions to use it problems will occur. Ultimately, one cannot administer an Enterprise Edition administration if one does not know how to administer the host Windows operating system as well as the DBMS being used to host the Enterprise server.
Database Administrator Edition

Database Administrator Edition

Manifold System **Database Administrator Edition** is a version of Manifold System that includes additional capabilities for setting up and managing GIS data storage in DBMS systems not found in other editions. If you have purchased and installed Database Administrator Edition the features described in this topic as well as in the Tools - Administrator Console topic and the Tools - Batch Export topic will be available to you. Database Administrator Edition is also included within **Ultimate Edition**. If you have installed some Manifold System edition other than Database Administrator or Ultimate editions you will not be able to use these features.

Database Administrator Edition is a superset of Enterprise Edition and includes all features found in **Enterprise Edition**. In addition, Database Administrator Edition includes a powerful managerial console called the Administrator Console that enables IT and DBMS administrators to configure data sources used to store the enterprise's geospatial data. The Administrator Console allows administrators to configure GIS data storage within the data source so that the data source is easier and more convenient to use for ordinary GIS users.

Database Administrator Edition also includes the Tools - Batch Export tool that facilitates rapid, mass uploading of Manifold drawings, images and surfaces into Oracle spatial databases. Upload of drawings requires an Oracle database server with Oracle SDO_GEOMETRY capability. Upload of images and surfaces requires an Oracle database server that supports Oracle GeoRaster capability. Oracle Spatial supports both types. Note that Oracle Express Edition supports SDO_GEOMETRY but not GeoRasters.

The intent of Database Administrator Edition is to make it easier for organizations to concentrate the expertise they have available for DBMS and IT administration, so that relatively few expert users can make it possible for relatively many, less expert users to take advantage of the power of geospatial data centralization within enterprise-class DBMS storage.

**Licensing**

Organizations that store their geospatial data within databases will acquire Database Administrator Edition licenses for their IT and DBMS administrators in addition to however many Enterprise Edition or Universal Edition licenses they acquire for other users. The administrator will use Database Administrator Edition to set up the DBMS for more convenient use by ordinary users, for example, by setting up friendly names or by enabling the storage of formatting for drawings as well as geometry and attributes.

Once Database Administrator Edition is used to configure a database with user-friendly features, any Manifold edition connecting to that database (such as Enterprise Edition or Universal Edition) can utilize those features. Manifold editions other than Database Administrator Edition that connect to the database will not be able to override or alter the settings specified using Database Administrator Edition. This assures that expert configuration by administrators will not be altered into chaos by inexpert users.

Although a typical large organization might have a few dozen Database Administrator Edition licenses with many hundreds of Enterprise Edition licenses, some organizations will have a higher ratio of Database Administrator Edition licenses if the organization has a higher percentage of users with the skills and administrative responsibilities to be trusted with administrator functions. For example, a very small organization such as a consultancy with a staff of five professionals, each of whom has great expertise and is trusted by his or her peers with DBMS administrative rights, might simply install Database Administrator Edition licenses for all five professionals. Each professional could then manage the DBMS as necessary, for example, uploading drawings and then right away configuring the data store for subsequent employment by his or her peers.

Conversely, an organization with relatively few administrators and many inexpert users will want to centralize all management of their data store within those few administrators and will therefore procure a relatively lower ratio of Database Administrator Edition licenses compared to the bulk of Enterprise Edition or Universal Edition licenses procured for ordinary users.

**Benefits for Manifold System Users**

Although the server statistics reporting capability of Administrator Console will prove useful in any environment, the client benefits of Database Administrator Edition such as friendly names will extend only to subsequent usage by Manifold System users and not to users of other GIS software. Legacy GIS software from other vendors does not in general have the capabilities of Manifold so the benefits of enhanced capability or user-friendly features made possible by Database Administrator Edition will not be available outside of Manifold System.
However, the client benefits of Database Administrator Edition are provided in a neutral way so that if a particular data storage method is used, such as SDO_GEOMETRY in Oracle Spatial, that supports multi-vendor interoperability then such interoperability will be preserved. Database Administrator Edition provides enhancements for the benefit of Manifold System users without interfering with least common denominator interoperability for users of other software.

In fact, Database Administrator Edition even allows DBMS administrators to identify some types of legacy storage, for example, using "open" binary geometry storage formats such as OGC WKB, as drawings that can be enrolled within Manifold user-friendly views. This allows Manifold users to still enjoy extended capabilities and user-friendly features even if the original data store was loaded from a legacy GIS system.

**Workflow**

DBMS administrators will normally employ Database Administrator Edition in the following workflow for each component uploaded to the data source in use:

1. Prepare the component for upload. For example, when exporting a drawing that is to be used for versioned edition to a non-Oracle DBMS we will have to create a version column. When exporting a drawing containing multiple variable-length columns to Oracle we will have to convert some of those columns to fixed length so that the drawing does not exceed the Oracle limit of one variable-length column per table.

2. If we are working with an Oracle database we can upload the component to the DBMS using the Tools - Batch Export command to export many components at once to Oracle. If the DBMS is not Oracle, we can export a drawing by exporting the drawing’s table using File - Export and including the Geom (I) column as is shown in the Storing Drawings in SQL Server example topic.

3. Using Administrator Console, specify a friendly name to be used for that component.

4. Using Administrator Console, enable formatting, if desired, for drawings.

5. Using Administrator Console, specify link and import options for the component, for example, such as the version column to be used for versioned editing.

Step 1 above is accomplished using Manifold tools such as capabilities in Tables that allow adding new columns and changing column types.

Step 2 above is normally accomplished using File - Export dialogs or the Batch Export tool if an Oracle database is being used.

Steps 3, 4 and 5 are accomplished using the Administrator Console that is provided within Database Administrator Edition.

See the Using Administrator Console topic for details and examples on how the above workflow is accomplished using the Administrator Console. See the Tools - Batch Export topic for details on using Batch Export.

In addition to the above cycle for each component uploaded, the administrator will of course still manage the host DBMS in the usual way using whatever tools are provided for that DBMS. For example, administrators will create databases, create user logins and assign user roles and so forth. Such activity is DBMS-specific and will be accomplished using the administrative tools provided by the DBMS vendor.

**See Also**

Database Installations  
Data Storage Strategies  
Multi-User Editing of Linked Drawings  
Linked Images from Oracle Servers  
Enterprise Edition  
Oracle Spatial Facilities  
Using Administrator Console  
Database Object Properties  
Database Object Triggers  
Storing Drawings in SQL Server  
Tools - Administrator Console
Tools - Batch Export
Tools - Database Console
Using Administrator Console

This topic describes a feature that is provided only within Manifold Database Administrator Edition and is not available in other Manifold System editions. If you have purchased and installed Database Administrator Edition the features described in this topic will be available to you. If you have installed some other Manifold System edition you will not be able to use these features.

The **Tools - Administrator Console** dialog provides a user interface for DBMS administrators to configure and manage Manifold geospatial data storage within enterprise DBMS data stores. The **Administrator Console** includes the following capabilities:

- Designate tables containing geometry columns to be treated as drawings by Manifold clients.
- Specify a generic spatial index to be used with tables containing geometry columns.
- Designation of user-friendly, **component** names for Manifold components stored within databases. These user-friendly component names can then appear within Database Console in all Manifold editions capable of connecting to that data source. Friendly names make it a lot easier for users to find their data within large databases when linking drawings or other components.
- Automatic creation and maintenance of an internal metadata table, called **MFD_META** by default, within the data source that is used to store housekeeping and metadata information on database objects for use by Manifold System.
- Displaying the number of objects in drawings and the dimensions and number of channels in images.
- Displaying the projections of drawings and images.
- Editing the coordinate system of a database object by double clicking the relevant cell in the **Projection** column.
- Enabling and disabling storing formatting of a database drawing object in the data source by double clicking the relevant cell in the **Format** column. When enabled, formatting is stored in the metadata table in XML format.
- Viewing and editing the location precision of a drawing database object via the **Precision** column.
- Viewing the number of records in tables and views via the **Statistics** column.
- Viewing triggers bound to a database object via the **Triggers** toolbar button. This capability is enabled only for Oracle native data sources.
- Editing the default link options of a database table object containing geometry data in one of the supported binary column geometry formats so it behaves like a drawing. This allows easy views of such database objects as drawings in subsequent Manifold sessions.
• Editing the default link options (geometry column and type, version column, server version increment option, formatting storage) and other properties (friendly name, coordinate system, precision) of a database object via the Properties toolbar command. This allows rapid and simplified linking in subsequent Manifold sessions.

Note: image capabilities apply if the data source in use supports images. Currently, only Oracle data sources with GeoRaster capability support linked image storage.

**Administrator Console Layout**

When connected to a data source the Administrator Console shows database objects in the data source in a contents pane. The lower pane is a command pane in which commands may be written using the command language of the data source, normally SQL.

Highlighting an object by clicking on it makes it the context object (that is, puts the focus on that object) so that toolbar buttons such as the Properties or Trigger button can operate on it. The status bar will also report information on the context object.

**Controls**

**Data Source** Choose the name of a data source, or press the browse button to launch the Connect To dialog to specify a data source. ADO.NET, ODBC, OLE DB and file-based data sources may also be used. The Server box will show the last used name or connection string (Manifold will not actually connect to the last used name or connection string until the Refresh button is pressed). Any PASSWORD and PWD parameters used in a connection string will be masked in the last used name or connection string to protect security.
(contents pane) Displays the contents of the data source.

**Refresh** - Update the contents pane.

**Properties** - Launch the Properties dialog for the highlighted database object. Allows specification of the friendly name, precision, version column, formatting and server version incrementing options. Also used to configure tables for treatment as drawings with specification of geometry columns and geometry data type.

**Triggers** - Launch the Triggers dialog for the highlighted database object. View and edit triggers bound to the database object (Oracle only). Shows the trigger, type of trigger and event for each trigger as well as the procedure text.

**Run** - Press to execute the command line text in the command pane.

**Global Filter** - Launches a dialog enabling filtering the names of database objects using regular expressions. Changes to the global filter are saved between different sessions of Manifold. By default, the global filter includes several masks suppressing display of system tables commonly found in Oracle databases. See the Tools - Database Console topic for use of the Global Filter button.

**Columns** - Toggle display of desired columns on or off in the contents pane. Manifold will remember the last used columns between sessions. Data is fetched for columns from the data source on demand.

Columns include:

- **Component** - Show the "friendly" name, that is, the Manifold component name.
- **Format** - Enable storage of formatting for drawings. Double-click into the Format cell of a database object to enable or disable formatting.
- **Precision** - Show location precision used in drawings.
- **Projection** - Projection used by this database object. Double-click into the Projection cell of a database object to launch the database object Projection dialog.
- **Spatial Index** - Reports if a generic spatial index is present for this object. Double-click into the Spatial Index cell of a database object to launch a dialog that enables creating and editing the spatial index. The bounding box of the area covered by the index is editable.
- **Statistics** - Useful information, such as the number of objects in drawings or the number of records in tables.
- **Triggers** - The number of database triggers, if any, for this object. Double-click into the Triggers cell of a database object to launch the database object Triggers dialog.

**Filter** Filter by character sequence - Show only those database objects in the contents pane the name of which contain the given sequence of characters. This is particularly helpful when working with data sources with lots of objects, such as SQL Server or Oracle data sources. See the Tools - Database Console topic for use of the Filter box.

(command pane) A text pane in which command lines, such as SQL, can be entered to be executed by the data source.

(status bar) Information about the highlighted object will be provided
at the bottom of the dialog.

Spatial Index Dialog Controls

The spatial index dialog appears when double-clicking into the Spatial Index cell for a database object that Manifold has been instructed to treat as a drawing.

Object
The database object to index. This will be a table name followed by a dot followed by the geometry column to be indexed. A typical name in SQL Server Express, for example, might be `dbo.Mexico Table.GeomI` if we exported a table using the `Geom (I)` intrinsic field to provide a geometry column.

Use spatial index
Check to create a spatial index.

X and Y boxes
Four boxes for the X and Y extents to be covered by the spatial index, initially blank until the Suggest button is pressed. The outer two boxes (not editable) show the extents of the data set. The inner two boxes show the extent of the spatial index to be constructed. This is normally the full extent of the data set, although in special cases users may wish to create a spatial index that covers less than the entire data set.

Level
Degree of detail in the spatial index, providing a balance between granularity and performance. Higher values produce more granular indices, which require more space but are potentially more efficient. For best efficiency, choose an index level proportional to the number of objects expected to be in the drawing. The value suggested by the Suggest button is a reasonable default.

Suggest
Press this button to command Manifold to analyze the data set to determine the extents of the data and a recommended

Designating a Friendly Name

To enter a friendly name for a database object, click on the object to highlight it and then double-click into the Component cell for that object or press the F2 key. Friendly names may include spaces and can be any component name legal within a Manifold project.

Using Friendly Names within the Database Console

To use friendly names within Database Console, connect to a data source that has a metadata table and friendly names designated with Administrator Console, and turn on component view using the Component View toolbar button. If the data source does not contain a metadata table set up with Administrator Console, the component view button will be disabled.

When in component view, the Database Console dialog will only show components that have been assigned friendly names. The Filter box will work with friendly names as well.

The MFD_META Table

DBMS administrators may be interested in additional information about the MFD_META metadata table that is created and used by the Administrator Console. Friendly names are stored in the metadata table and are available to all Manifold clients connecting to the data source.

The metadata table is named `MFD_META` by default and contains three fixed-length or variable-length text columns named OBJ, PROP and VAL. The OBJ column stores names of database objects, the PROP column stores property names, and the VAL object stores property values. If there is both a system table named `MFD_META` and a user table named the same (as might be the case with Oracle data sources), the system table is given preference.
Supported properties are:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component</td>
<td>Friendly name. Can include spaces, punctuation and other characters allowed in a Manifold component name.</td>
</tr>
<tr>
<td>CoordinateSystem</td>
<td>The coordinate system of a table behaving as a drawing, in Manifold XML format.</td>
</tr>
<tr>
<td>Epsilon</td>
<td>The location precision parameter of a table behaving as a drawing, in native drawing units.</td>
</tr>
<tr>
<td>Format</td>
<td>True or False, depending on whether or not a drawing should store formatting on the data source.</td>
</tr>
<tr>
<td>FormatAreaBack</td>
<td>Area background color format for a drawing, in Manifold XML format.</td>
</tr>
<tr>
<td>FormatAreaBorderBack</td>
<td>Area border background color format for a drawing, in Manifold XML format.</td>
</tr>
<tr>
<td>FormatAreaBorderFore</td>
<td>Area border foreground color format for a drawing, in Manifold XML format.</td>
</tr>
<tr>
<td>FormatAreaBorderSize</td>
<td>Area border size format for a drawing, in Manifold XML format.</td>
</tr>
<tr>
<td>FormatAreaBorderStyle</td>
<td>Area border style format for a drawing, in Manifold XML format.</td>
</tr>
<tr>
<td>FormatAreaFore</td>
<td>Area foreground color format for a drawing, in Manifold XML format.</td>
</tr>
<tr>
<td>FormatAreaSize</td>
<td>Area size format for a drawing, in Manifold XML format.</td>
</tr>
<tr>
<td>FormatAreaStyle</td>
<td>Area style format for a drawing, in Manifold XML format.</td>
</tr>
<tr>
<td>FormatLineBack</td>
<td>Line background color format for a drawing, in Manifold XML format.</td>
</tr>
<tr>
<td>FormatLineFore</td>
<td>Line foreground color format for a drawing, in Manifold XML format.</td>
</tr>
<tr>
<td>FormatLineSize</td>
<td>Line size format for a drawing, in Manifold XML format.</td>
</tr>
<tr>
<td>FormatLineStyle</td>
<td>Line style format for a drawing, in Manifold XML format.</td>
</tr>
<tr>
<td>FormatPointBack</td>
<td>Point background color format for a drawing, in Manifold XML format.</td>
</tr>
<tr>
<td>FormatPointFore</td>
<td>Point foreground color format for a drawing, in Manifold XML format.</td>
</tr>
<tr>
<td>FormatPointRotation</td>
<td>Point rotation format for a drawing, in Manifold XML format.</td>
</tr>
<tr>
<td>FormatPointSize</td>
<td>Point size format for a drawing, in Manifold XML format.</td>
</tr>
<tr>
<td>FormatPointStyle</td>
<td>Point style format for a drawing, in Manifold XML format.</td>
</tr>
<tr>
<td>GeometryColumn</td>
<td>Name of a column to use for geometry data. If present, makes the table behave as a drawing.</td>
</tr>
<tr>
<td>GeometryColumnType</td>
<td>Data type of the column used to store geometry data. Can be Manifold for Geom, SDE for GeomSDE, SHP for GeomSHP, or WKB for GeomWKB. If absent or invalid, the data type is assumed to be Geom.</td>
</tr>
<tr>
<td>VersionColumn</td>
<td>Name of a column to use for version data, for drawings.</td>
</tr>
<tr>
<td>VersionColumnAuto</td>
<td>True or False depending on whether or not Manifold should rely on the server to increment version data after making changes to the metric of drawing objects.</td>
</tr>
</tbody>
</table>
Tech Tip

Almost always after specifying that a table containing geometry be interpreted as a drawing we will click into the Spatial Index column for that table object in Administrator Console to specify a generic spatial index for that table. Specifying a spatial index will enable Manifold clients importing or linking that table as a drawing to work with only the area of interest desired. This is potentially a huge performance enhancement when working with very large drawings.

Keep in mind that the choice of connection technology will influence how Manifold interacts with the DBMS. For example, ADO .NET connections typically are read-only and will not allow either Administrator Console or Database Console to make changes, such as adding a spatial index, to the database. In another example, connecting to a spatial DBMS using the native connection technology (such as using OCI to connect to Oracle) tells Manifold we want to use the DBMS’s native spatial technology, while using ODBC or OLE DB tells Manifold to connect in a generic way. See the Spatial DBMS Facilities topic for additional information.

See Also

Database Administrator Edition
Database Object Properties
Database Object Triggers
Multi-User Editing of Linked Drawings
Storing Drawings in SQL Server
Tools - Administrator Console
Tools - Database Console
View - Panes - Review
Database Object Projection

This topic describes a feature that is provided only within Manifold Database Administrator Edition and is not available in other Manifold System editions. If you have purchased and installed Database Administrator Edition the features described in this topic will be available to you. If you have installed some other Manifold System edition you will not be able to use these features.

Launch the database object Projection dialog within Administrator Console by double-clicking into the Projection column cell for a database object or by highlighting the object, clicking the Properties toolbar button and then clicking the [...] browse button for Projection in the Properties dialog.

The database object Projection dialog has two uses:

- For components exported to Oracle databases from Manifold System it allows changing the Oracle coordinate system in use. The usual course of events is that when a component is exported from Manifold into Oracle the coordinate system assigned is matched and, if necessary, altered in the export dialog. The database object Projection dialog in the Administrator Console allows changing this projection.

- The second use of the database object Projection dialog is to assign a projection to a drawing stored in a non-Oracle database in a table using a geometry format such as WKB that does not save coordinate system information. In that case, the Assign Projection dialog is raised and used as it would be to assign the projection to a drawing in Manifold that was imported from a GIS format that does not store projection information.

The illustration above shows the database object Projection dialog for a drawing stored in Oracle. The controls specified below are for that Oracle form of the database object Projection dialog.

**Controls**

- **Original**
  - Original coordinate system used within Manifold.

- **Map to**
  - A list of Oracle coordinate systems with the SRID and Name for each. The coordinate system in use is highlighted.

  (description pane) Detailed description of the Oracle coordinate system.
Comments, for example, noting that a given Oracle coordinate system is exactly the same as the original Manifold coordinate system.

**Auto**  Automatically choose the Oracle coordinate system that is the best match to the original Manifold coordinate system.

### See Also

- Database Administrator Edition
- Database Object Properties
- Database Object Triggers
- Edit - Assign Projection
- Example: Storing a Drawing in Oracle
- Oracle Spatial Facilities
- Storing Drawings in SQL Server
- Tools - Administrator Console
- Tools - Database Console
- Using Administrator Console
Database Object Properties

This topic describes a feature that is provided only within Manifold Database Administrator Edition and is not available in other Manifold System editions. If you have purchased and installed Database Administrator Edition the features described in this topic will be available to you. If you have installed some other Manifold System edition you will not be able to use these features.

Highlight a database object and click the **Properties** toolbar button in the Administrator Console dialog to launch the **Properties** dialog for that object.

The **Properties** dialog has the following uses:

- For drawings, the **Properties** dialog allows setting default options for subsequent import or linking back into Manifold projects. Settings made via the **Properties** dialog will be automatically incorporated when users later employ **Database Console** to import or link the component into their projects. This eliminates the need for ordinary users to know how to handle various options.

- For tables or queries which incorporate geometry data in the form of binary columns, the **Properties** dialog allows use of such database objects as drawings within Manifold. The Manifold database administrator can use the **Properties** dialog to designate such tables as drawings and to set forth which columns in that table should be treated as containing geometry and what type of geometry format is used.

- For images, the **Properties** dialog allows setting basic properties such as the coordinate system and the friendly name. Manifold works with images via Oracle GeoRaster technology, which requires the use of Enterprise Edition and an Oracle product that supports GeoRasters, such as Oracle Spatial (Oracle Express does not support GeoRasters). See the Oracle Spatial Facilities topic for details.

### Controls

- **Object** Context database object (item that was highlighted when the **Properties** button was clicked).

- **Name** Manifold component name, also referred to as the friendly name.

- **Treat as drawing** Checked by default (and disabled) for all Manifold drawings uploaded by Manifold into the database. Should be checked for tables that contain geometry data that are to be treated by Manifold as drawings.
Projection  Projection used by the component. Click the [...] browse button to specify the database projection to be used for this object via the database object Projection dialog.

Precision  Location precision used for this drawing.

Geometry  Name of the column used to store geometry. Filled in by default for drawings uploaded by Manifold.

Type  Type of geometry storage used in the geometry column, one of the geometry storage types usable by Manifold (see the Geometry in Tables topic). The Manifold type, Geometry, is automatically mapped in the case of Oracle into SDO_GEOMETRY so that when Geometry appears in this box for Oracle tables we know that the actual storage is SDO_GEOMETRY.

Version  The column to use for versioned editing in concurrent multi-user editing scenarios. See the Multi-User Editing of Linked Drawings topic. The table must contain a primary key to allow a version column to be designated.

Rely on server to increment version automatically  Use a trigger in the server to automatically increment the version column to support versioned editing. The default setting for Oracle.

Store formatting on server  Enable storage of drawing formatting within the database. Uses the metadata table created and managed by Administrator Console to store formatting.

Example

This example assumes we have uploaded a table that contains geometry data in OGC WKB format. We will use the database object Properties dialog to configure this table for use as a drawing within Manifold.

This is something of an artificial example using manual creation of geometry and manual upload and configuration via the Properties dialog. Exporting a drawing to SQL Server using default settings will automatically configure the drawing in the database for subsequent importing or linking into Manifold.

However, it could be the case that if we are using non-Manifold sources of geometry data (such as some other application that writes WKB data) we will need to configure it for use as a drawing within Manifold. This example shows how to do that.

Step 1: Arrange Example Data

Before we can get to the main part of the example, showing how to work with geometry data in tables within a database, let's create some example data and put it into a database. We will deliberately use a highly manual way of creating geometry data using an "open" geometry type, WKB, to simulate a situation where some third party application may have written geometry data to our SQL Server database.

```
<table>
<thead>
<tr>
<th>Name</th>
<th>geom geom</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minnesota</td>
<td>&lt;geom wkb, polygon&gt;</td>
<td>0</td>
</tr>
<tr>
<td>Washington</td>
<td>&lt;geom wkb, multipolygon&gt;</td>
<td>0</td>
</tr>
<tr>
<td>Idaho</td>
<td>&lt;geom wkb, polygon&gt;</td>
<td>0</td>
</tr>
<tr>
<td>Montana</td>
<td>&lt;geom wkb, polygon&gt;</td>
<td>0</td>
</tr>
<tr>
<td>North Dakota</td>
<td>&lt;geom wkb, polygon&gt;</td>
<td>0</td>
</tr>
<tr>
<td>Michigan</td>
<td>&lt;geom wkb, multipolygon&gt;</td>
<td>0</td>
</tr>
<tr>
<td>Maine</td>
<td>&lt;geom wkb, multipolygon&gt;</td>
<td>0</td>
</tr>
<tr>
<td>Wisconsin</td>
<td>&lt;geom wkb, multipolygon&gt;</td>
<td>0</td>
</tr>
<tr>
<td>Oregon</td>
<td>&lt;geom wkb, polygon&gt;</td>
<td>0</td>
</tr>
<tr>
<td>South Dakota</td>
<td>&lt;geom wkb, polygon&gt;</td>
<td>0</td>
</tr>
</tbody>
</table>
```
Consider a table that contains geometry information in OGC WKB form within a column called `wkbgeom`. The table has had a `Version` column added to it suitable for use with versioned editing if this table will at some point be treated as a drawing. The `Version` column is simply a long integer type.

**Note:** For the sake of this example, the above table was created within Manifold beginning with the US_Main sample drawing as follows: First, the drawing was imported into Manifold. Next, the drawing’s table was opened and a new field called `wkbgeom` was added to the table as a Geometry (WKB) type. The transform toolbar was then used to copy the contents of the Geom (I) intrinsic field to the `wkbgeom` field, with Manifold automatically converting the `geom` type into the WKB geometry type on the fly. Finally, we deleted all columns except those seen above and we added a `Version` column.

The result is a table like that created by some other software package we might encounter in a database. For the sake of this example, we export the table into a SQL Server 2005 Express Edition database using an OLE DB connection in the Data Source dialog.

**Step 2: Launch Administrator Console Properties Dialog**

We connect to the SQL Server database with Administrator Console and see the database object representing the table uploaded in the previous step.

**Administrator Console** shows the database object as a table. We can click on it to highlight it and then press the Properties toolbar button to launch the Properties dialog.

**Note:** If we had uploaded the table into an Oracle database, Manifold would have converted the geometry column automatically into native Oracle SDO_GEOMETRY. Connecting to the Oracle database with Administrator Console would then automatically have shown the table as a drawing. We deliberately used SQL Server in this example so that the automatic conversion would not be done, therefore allowing us to show in this example how to use Administrator Console to specify treatment of a table as a drawing.
The **Properties** dialog shows the table with no other information initially. We will first give it a component name and check the box for it to be treated as a drawing.

We provide a friendly, component name and check the **Treat as drawing** box. When we check the box Manifold will enable the other controls used with drawings. We will begin by clicking the [...] browse button to specify the **Projection** to be used.
Manifold launches the Assign Projection dialog using default Orthographic projection. If this data is not intended to represent data in Orthographic projection, now is the time for us to specify the coordinate system that it uses.
Because we created this table for this example from the sample US Main drawing, we know that the data uses Latitude / Longitude and WGS84 datum and we can set the Assign Projection dialog accordingly. If we did not know that, there is nothing about the OGC WKB format for geometry that would have told us that. Unlike sensible geometry types for databases, such as Oracle spatial types or Manifold’s Geom type, the WKB format created by the Open GIS Consortium is too stupid to represent GIS notions such as the coordinate system to be used for data. [Chalk up another example for why the phrase “design by committee” is used as pejorative.]

Press OK to close the Assign Projection dialog and assign the new projection.
We can see that now the Projection is reported as Latitude / Longitude. We note that the Precision value is still empty, so we'll have to fill that in.

The Geometry column has already been correctly pre-loaded with the wkbgeom column name. If there were more than one binary data column in the table Manifold would not know that the binary data wkbgeom column is the right one, but since it is the only binary column type in the table and we've already told Manifold this table is to be treated as a drawing, Manifold knows it is a safe bet that the only binary column in the table is the one to guess we intend to use for the Geometry column. However, we have to tell Manifold what Type of geometry data is in that column. Manifold has loaded the box with Geometry as the type, that choice being simply the first choice in alphabetical order.

We specify the Precision value (another magic number we know from our experience with the US Main drawing that otherwise would be impossible to know).

For the Type column we choose Geometry (WKB), the data type used for OGC WKB data.
Finally, we choose Version for the version column and check the Store formatting on server box and press OK.

These last two settings are Manifold-specific choices, since only Manifold can use the Version column for versioned, concurrent multi-user editing and only Manifold will be using any formatting used for this drawing from Manifold clients. For the sake of completeness we show these choices.

As a practical matter, it is slightly goofy to use WKB as a storage format for drawings that will be used exclusively with Manifold. A much better choice is Manifold Geometry type for storage within SQL Server or DB2. Better still, use SDO_GEOMETRY within Oracle (effectively the same thing, since with Oracle Manifold Geometry is automatically converted to and from SDO_GEOMETRY on the fly).

Using Manifold Geometry within SQL Server is great, but does not allow interoperability with old-fashioned GIS applications. If fast and modern interoperability is the goal, using Oracle SDO_GEOMETRY delivers it. Hundreds of GIS applications can interoperate with Oracle SDO_GEOMETRY, so if one wants to use an "open" format that is sensibly designed for geometry storage SDO_GEOMETRY as a practical matter is just the ticket.

WKB is used only when we must retain the ability to interoperate with old-fashioned GIS software using non-Oracle DBMS servers. But, at least it's there for those users who get forced into such things, hence this example.
Back in Administrator Console the display will be immediately updated with the new information on the database object. If we press Refresh we will also see, as in the illustration above, that a new table has appeared in the database, the MFD_META table maintained by Administrator Console in support of Manifold features such as component names (friendly names) and formatting.

If some other Manifold user used Database Console to connect to that SQL Server database he or she would see a drawing listed under the friendly, component name we assigned using Administrator Console.
Users can simply click on the drawing and press **Link** to immediately link the drawing into the project.

The drawing is immediately linked without any need for intermediate dialogs, since **Administrator Console** was already used to specify desired import and link options. For example, the **Version** column is automatically specified to enable versioned editing with no need for users to remember to specify a version column.

Double-clicking the drawing in the project pane opens it and shows that it does capture the **US Main** example drawing within the **WKB** geometry stored in the SQL Server table.
We can format the drawing as seen above, assigning a thematic format to area background color using the **Area** (I) intrinsic field, so states are colored in natural breaks using their sizes. Since storage of formatting has been automatically turned on for us by the settings made in **Administrator Console**, whatever formatting we apply will persist on the server and will be available the next time we link the drawing into another Manifold project.

**Tech Tip**

Almost always after specifying that a table containing geometry be interpreted as a drawing we will click into the **Spatial Index** column for that table object in **Administrator Console** to specify a generic spatial index for that table. Specifying a spatial index will enable Manifold clients importing or linking that table as a drawing to work with only the area of interest desired. This is potentially a huge performance enhancement when working with very large drawings.

**See Also**

- Database Administrator Edition
- Database Object Projection
- Database Object Triggers
- Tools - Administrator Console
- Tools - Database Console
- Using Administrator Console
Database Object Triggers

This topic describes a feature that is provided only within Manifold Database Administrator Edition and is not available in other Manifold System editions. If you have purchased and installed Database Administrator Edition the features described in this topic will be available to you. If you have installed some other Manifold System edition you will not be able to use these features.

Triggers are procedures that run implicitly before or after certain actions, as specified by the database administrator. At present, Manifold shows triggers only in native Oracle data sources.

The languages that may be used to write triggers vary from one database to another. In Oracle we can write actions in PL/SQL, Java or C while in SQL Server, we can write actions in T-SQL or .NET languages, such as VB .NET and C#.

The actions to which triggers can be bound also vary between databases. Most databases can schedule triggers to run before or after inserting, updating or deleting records in a particular table. Many databases support binding triggers to numerous other actions. Some actions can be initiated by users, and some by the database itself.

In some cases Manifold provides options to automatically create triggers to support system functionality. For example, when exporting a drawing to a native Oracle data source Manifold provides an option to automatically create a trigger to populate ID column values for newly inserted records, and another trigger to update version column values in changed records. We would normally choose to create such triggers but sometimes this will be unnecessary or undesired due to database design policies or for other reasons.

![Triggers dialog]

Highlighting a database object in Administrator Console and pressing the Triggers toolbar button launches the Triggers dialog to show any triggers associated with that object. Double-clicking into the Triggers column cell for that database object will also launch the Triggers dialog.

Controls

- (object name) Context database object (item that was highlighted when the Properties button was clicked).
- (triggers pane) A list of triggers for this object, showing for each trigger the name, type of trigger and type of event.
- (procedure pane) The procedure for the trigger highlighted in the triggers pane.
The *Triggers* dialog will show all triggers, not just those added by Manifold. It is therefore a handy tool to examine database objects to help troubleshoot complex databases that may include numerous objects unrelated to Manifold operations or to identify unwanted triggers added in error to Manifold objects.

**Example**

The UPDATE trigger written by Manifold to enable automatic server side incrementing of the *Version* column to support versioned editing is a classic example of a trigger.

To see such an example, connect to an Oracle database that contains a drawing for which the *Rely on server to increment version automatically* option has been enabled, click on the drawing to highlight it in *Administrator Console* and press the *Triggers* toolbar button.

Click on the UPDATE trigger in the *Triggers* dialog to see the procedure for that trigger in the procedures pane.

**See Also**

Database Administrator Edition
Database Object Properties
Tools - Administrator Console
Tools - Database Console
Using Administrator Console
Spatial DBMS Facilities

Manifold includes extensive features to take advantage of spatial DBMS capabilities, even with those DBMS products that do not provide native spatial DBMS capability. Before continuing with this topic, please see the Spatial DBMS topic for a general introduction to spatial DBMS and Manifold.

Manifold Enterprise Edition or above is required to connect using native connections to spatial DBMS such as Oracle Spatial, DB2 with IBM Spatial Extender, SQL Server 2008 spatial and PostgreSQL / PostGIS.

Storing Drawings within Spatial DBMS

Manifold features integrated read / write / edit, multiuser connections to a variety of spatial DBMS packages. In general, operation is fully automatic: when a Manifold drawing is uploaded to a spatial DBMS the Manifold geometry will automatically be converted into the native DBMS geometry type. When a drawing is linked or imported from a spatial DBMS source the native DBMS geometry will be automatically cast into Manifold geometry.

Manifold provides two ways of storing drawings within spatial DBMS:

- **Native storage within a spatial DBMS** - Examples are Oracle Spatial, IBM DB2, Microsoft SQL Server 2008 or PostgreSQL using the native geometry type specified by the DBMS. Connecting to one of these spatial DBMS packages using the "native" connection, such as OCI for Oracle, tells Manifold to automatically convert local Manifold geometry for drawings into whatever the target spatial DBMS users as its own native geometry type.

- **Non-native storage within almost any DBMS** - Manifold can confer spatial DBMS capability to almost any DBMS. When using Manifold-managed spatial storage, we have the choice of what geometry type we would like to use. Examples are storing drawings as tables within, say, MySQL or SQL Server using any of the binary geometry types supported by Manifold, such as GeomWKB (OGC WKB), GeomSHP (legacy ESRI geodatabases), or GEOMETRY (Manifold's own high performance geometry type, often abbreviated GEOM). Normally the choice will be made between Manifold GEOM type or OGC WKB type. In addition, Manifold can establish a generic spatial index. The spatial index is used together with the binary data to allow "ordinary" DBMS packages to function as a spatial DBMS. A special case of Manifold-managed storage is the use of the Manifold Spatial Extender for SQL Server 2005 to provide especially fast spatial DBMS capability within SQL Server 2005. See the Example: Storing a Drawing in Manifold Spatial DBMS topic.

It is quite possible to use any of the above storage methods within the same DBMS. For example, Oracle provides Locator spatial capability so drawings can be stored in Oracle using Oracle's native SDO_GEOMETRY type as with a typical spatial DBMS. But Oracle also provides a full roster of other DBMS types, including generic binary storage, which may be used for non-native geometry types. If desired, we can store non-native geometry within Oracle tables, establish generic indices and so on. In fact, within the same Oracle Spatial DBMS Manifold could have three different drawings stored as three different tables with each table using a different geometry type.

Storing Images and Surfaces within Spatial DBMS

Manifold provides two ways of storing images and surfaces within a DBMS:

- **Oracle GeoRaster** - Connecting via OCI (Oracle's native Oracle Call Interface) to an Oracle database that has GeoRaster capability (Oracle Spatial or Oracle Enterprise) allows us to store images within Oracle's native GeoRaster data type. See the Example: Storing an Image in Oracle topic.

- **Generic image storage** - Connecting via ODBC or OLE DB to almost any DBMS that allows binary storage allows us to store images and surfaces (which are stored as images with an additional Height channel). Images are stored as tiles, with the export process allowing choice of a tile size. Exporting an image to such a data source creates a table with a record for each image tile. Image tiles are stored in compressed or uncompressed BGR(A) format. Importing or linking images from such a data source automatically reassembles tiles. Surfaces cannot be linked, but can be imported by importing the Height channel of the image that is stored for them into the database. See the Example: Storing an Image in Manifold Spatial DBMS and Example: Storing a Surface in Manifold Spatial DBMS topics.

Manifold uses Database Console to connect to a database storing images to import or link to a stored image. Database console will recognize tables containing image tiles and will display them as images, allowing us to
import or link to the image by using the database console's Import or Link buttons. We can also import individual channels of such images as surfaces.

**Specifying the Spatial DBMS Technology to Use**

If Manifold can work with spatial DBMS using either a vendor's native spatial DBMS technology or spatial DBMS technology provided by Manifold, how do we specify which technology we would like to use for any particular drawing?

- **Choice of Type in Export Drawing Dialog** - When exporting a drawing to a database the Export dialogs give us a Type box which allows us to specify either Manifold, for Manifold-managed geometry types or, if a native connection technology was used to connect to a spatial DBMS, the name of that spatial DBMS vendor. Note that we must use the native spatial DBMS connection (see below) in the Data Source dialog to enable use of native spatial DBMS capabilities.

- **Automatic Choice when Importing or Linking** - Manifold dialogs for importing or linking will automatically use whatever technology is used for storage of that component. When Database Console connects to a data source Manifold will examine the source and determine what technology is in use and use that technology when importing or linking any given component.

For example, if we connect to an Oracle server when exporting a drawing we will see two choices available in the Type box: Manifold, and Oracle. Choosing Oracle will export the drawing into Oracle's native spatial system using SDO_GEOMETRY. Choosing Manifold will export the drawing using whatever non-native geometry type we specify in the dialog together with creation of a Manifold-established spatial index.

When importing or linking from a database if we want to use a spatial DBMS's native spatial facilities we must connect using that DBMS's native connection technology. Once connected, Manifold will automatically recognize whether the data in use is the vendor's native spatial geometry, Manifold-managed non-native spatial geometry or ESRI SDE geodatabase or Personal geodatabase geometry (see below for information on ESRI geodatabases).

**Data Source Connections for Spatial DBMS**

In addition to the usual roster of ADO.NET, OLE DB, and ODBC connections Manifold Enterprise Edition and higher editions provide specialized native DBMS connection choices that are used when adding a new data source using a spatial DBMS in the Data Source dialog:

<table>
<thead>
<tr>
<th>ADI.NET Data Sources</th>
<th>OLE DB Data Sources</th>
<th>ODBC Data Sources</th>
<th>PostgreSQL Data Sources</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2 Data Sources</td>
<td>Oracle Data Sources</td>
<td>SQL Server Data Sources</td>
<td></td>
</tr>
</tbody>
</table>

- **DB2 Data Sources** - Connect to IBM DB2 equipped with the DB2 Spatial Extender using IBM's native spatial connection technology.

- **Oracle Data Sources** - Connect to Oracle using OCI, the Oracle Call Interface to utilize Locator or Oracle Spatial facilities.

- **PostgreSQL Data Sources** - Connect to PostgreSQL equipped with the PostGIS spatial extender using native PostgreSQL connection technology. Connections made using this data source will have any password required masked so that later usage will not inadvertently expose the password.

- **SQL Server Data Sources** - Connect to Microsoft SQL Server 2008 spatial DBMS facilities using native SQL Server 2008 spatial connection technology.
Users sometimes ask if there is a special connection when using SQL Server 2005 as a spatial DBMS with the Manifold Spatial Extender for SQL Server installed. There is no need for a special connection because Manifold when connecting to SQL Server using any OLE DB connection will recognize and automatically use spatial indices managed by the spatial extender when the Manifold spatial extender has been installed.

**When Enterprise Edition is Required**

Manifold Enterprise Edition or above is required to connect using native connections to spatial DBMS such as Oracle Spatial, DB2 with IBM Spatial Extender, SQL Server 2008 spatial and PostgreSQL. Therefore, any usage of native spatial capabilities within these spatial DBMS products requires Enterprise Edition or above.

Manifold Enterprise Edition or above is required to create a spatial index when using Manifold to confer generic spatial DBMS capability upon almost any DBMS; however, once Enterprise Edition is used to create the spatial index any Manifold edition may be used to work with that data and that generic spatial index.

For example, we could use an Enterprise Edition license to load a SQL Server 2005 database with drawings supported with spatial indices, images and surfaces. We could then have many of our colleagues work with those drawings (including full read/write/edit capabilities), images or surfaces using Personal Edition licenses. However, if any of our colleagues wanted to upload a new drawing into that SQL Server database they would have to use an Enterprise Edition license to upload the drawing with an associated spatial index.

As a practical matter, large organizations that are working seriously with spatial DBMS using a natively spatial DBMS like Oracle Spatial or SQL Server 2008 will deploy Manifold using Enterprise licenses. Because of the very low cost of Enterprise Edition even those organizations that will use non-native spatial DBMS will tend to deploy Manifold using exclusively Enterprise Edition licenses so that all users can upload new drawings at any time.

In some specialized settings it can be useful to employ Personal or Professional licenses to connect to spatial DBMS to work with drawings previously uploaded using Enterprise Edition. Examples include small organizations on a budget, custom applications that utilize a Professional Runtime license, or organizations where there are many users with very undemanding, mainly "view only" interactions who will never be creating new drawings.

**Supported Spatial DBMS**

Current spatial DBMS products with spatial types understood by Manifold include:

- **IBM DB2 with Spatial Extender** - Spatial DBMS functionality with DB2 depends upon installation of the IBM Spatial Extender for DB2.
- **Microsoft SQL Server 2005** - Implemented using the Manifold Spatial Extender for SQL Server or through use of Manifold-managed generic spatial DBMS capability. See the SQL Server Spatial DBMS Facilities topic.
- **Microsoft SQL Server 2008** - Microsoft has engineered spatial DBMS functionality into SQL Server 2008. Manifold supports SQL Server 2008 spatial extensions today and has since the very first SQL Server 2008 spatial pre-release in 2007. See the SQL Server Spatial DBMS Facilities topic.
- **Oracle Spatial and Oracle Locator** - All Oracle products now have Locator capability to allow spatial DBMS capabilities with vector drawings. Oracle Spatial is required to store images and surfaces using Oracle's GeoRaster type. See the Oracle Spatial Facilities topic for additional information.
- **PostgreSQL** - An open source DBMS that is extended into a spatial DBMS via the PostGIS package.
- **ESRI SDE and Personal Geodatabases** - Not really a separate spatial DBMS but rather a collection of proprietary ESRI types, metadata tables and facilities imposed upon one of the DBMSs supported by ESRI. Since many ESRI people think of these as spatial DBMS they are enumerated here because Manifold can interact with them as well. See the ESRI Geodatabases topic.

In addition to the usual roster of ADO .NET, OLE DB, and ODBC connections Manifold Enterprise Edition and higher editions provide specialized native DBMS connection choices that are used when adding a new data source using a spatial DBMS in the Data Source dialog. See the Data Source dialog topic for a current list.

For specific discussion of different ways to use SQL Server in spatial DBMS, see the SQL Server Spatial DBMS Facilities topic.

**Notes on Specific Spatial DBMS Products.**
Oracle and SQL Server have dedicated topics discussing operations with those DBMS products. See the Oracle Spatial Facilities and the SQL Server Spatial DBMS Facilities topics.

**IBM DB2**

When IBM's DB2 Spatial Extender has been install we can use DB2 as a spatial DBMS, taking advantage of native DB2 geometry. The IBM Spatial Extender for DB2 is a free download for DB2 Express-C and other DB2 editions.

Manifold usage of DB2 spatial DBMS data sources uses Database Console as the primary interface and includes:

- Connecting to an DB2 data source via the native DB2 interface. The native DB2 interface is faster and more robust than accessing DB2 through intermediate database layers such as ODBC or OLE DB. Manifold will remember the last DB2 data source to which a successful connection has been made. Omitting a login name when connecting to a native DB2 data source uses integrated security.
- Importing, linking and exporting tables.
- Listing the drawings in an DB2 data source in Database Console.
- Importing drawings.
- Linking drawings in read-write mode.
- Exporting drawings.

When importing, linking or exporting drawings, reading data will automatically and transparently translate DB2 geometry into Manifold Geom values. Writing data will automatically and transparently translate Manifold Geom values into DB2 Geometry.

When linking a drawing or table from a native DB2 data source Manifold will examine the DB2 table privileges. If the detected table privileges do not allow writing the resulting component within Manifold will be read-only.

To work with DB2 data sources we must have all DB2 client DLLs (for example, DB2CLI.DLL plus any DLLs it depends on) to be installed in a folder referenced in the user's or system's PATH variable. The easiest way to be sure this is done is include the folder containing all DB2 client DLLs in the system PATH variable.

**PostgreSQL / PostGIS**

PostgreSQL (pronounced "post-gres-que-ell" in English) is an open source DBMS that has been extended into spatial capabilities via the PostGIS set of open source extensions.

Manifold usage of PostgreSQL data sources uses Database Console as the primary interface and includes:

- Connecting to a PostgreSQL data source via the native PostgreSQL interface. Manifold will remember the last PostgreSQL data source to which a successful connection has been made.
- Importing, linking and exporting tables.
- Listing the drawings in a PostgreSQL data source in Database Console.
- Importing drawings.
- Linking drawings in read-write mode.
- Exporting drawings. When exporting a drawing to a PostgreSQL data source there is an option to create a spatial index (on by default), and an option to create a sequence and an update trigger (on by default), similar to the options provided for export to Oracle. See the Example: Storing a Drawing in Oracle and Export Drawing - Oracle topics.

To work with PostgreSQL data sources we must have all PostgreSQL client DLLs (for example, LIBPQ.DLL plus any DLLs it depends on) to be installed in a folder referenced in the user's or system's PATH variable. The easiest way to be sure this is done is include the folder containing all PostgreSQL client DLLs in the system PATH variable.
When importing, linking or exporting drawings, reading data will automatically and transparently translate PostgreSQL geometry into Manifold Geom values. Writing data will automatically and transparently translate Manifold Geom values into PostgreSQL Geometry.

Exchanging geometry data with PostgreSQL tolerates data with Z and M values, although Z and M values are not used by Manifold. Reading geometry data with Z and M values throws these values away. Writing geometry data with Z and M values inserts zeros.

**ESRI Geodatabases**

Not really a spatial DBMS in the sense of the above DBMS products, ESRI SDE and Personal geodatabases are discussed in the ESRI Geodatabases topic.

**Deleting Drawings stored within Spatial DBMS**

Deleting a drawing stored within a spatial DBMS is usually simple, but requires a few nuances due to the multi-user / multi-process nature of most DBMS servers. See the discussion in the Tech Tip in the Database Console topic.

**Projections and Spatial DBMS**

Given the different methods of storing drawings within a DBMS there is some variation as to how projection information is also stored. In general, when importing or linking drawings from spatial DBMS Manifold will acquire the correct projection to use for that drawing. For the sake of discussing projections, we can consider three types of drawings stored in DBMS:

- **Native drawings**, that is, drawings which use geometry types and metadata or projection codes specific to a particular spatial database. Examples include drawings stored as SDO_GEOMETRY within Oracle, ST_GEOMETRY within IBM DB2 with Spatial Extender, native PostgreSQL geometry or native SQL Server 2008 geometry.

- **ESRI drawings**, that is, drawings which use geometry types and metadata specific to a particular ESRI technology, which may or may not be tied to a particular database. Examples include SDE drawings stored within SQL Server using ArcSDE or ESRI Personal Geodatabases stored within Access .MDB. See the ESRI Geodatabases topic for information on using such geodatabases with Manifold.

- **Manifold drawings**, that is, drawings stored by Manifold within any DBMS using Manifold's generic spatial DBMS capability (or using the Manifold spatial extender with SQL Server), regardless of which geometry type is used. Examples include drawings stored by Manifold using OGC WKB within MySQL with a spatial index created by Manifold or drawings stored by Manifold using Manifold Geom within, say, SQL Server Express with a spatial index created by Manifold.

**Native drawings** store coordinate systems using means specific to that particular spatial database. Such means usually include a table or a set of tables which store the definitions of all coordinate systems together with a table storing the bindings between columns in the database tables which contain geometry data and their coordinate systems.

Manifold can handle native drawings within DB2, Oracle and PostgreSQL and SQL Server 2008 spatial. Exporting a native drawing from Manifold allows selecting a coordinate system from the list of those registered in the database and creates the binding between the geometry column in the exported table and that coordinate system. Manifold will attempt to match the coordinate system in use to a coordinate system supported by the spatial DBMS. If an exact match is not found, Manifold will re-project the data into the closest coordinate system (projection) found. Importing or linking a native drawing into Manifold reads the binding and retrieves the coordinate system.

For projections within SQL Server 2008 native GEOMETRY type, see the comments on SQL Server 2008 projections in the SQL Server Spatial DBMS Facilities topic.

**ESRI drawings** are similar to native drawings but are not tied to a particular type of a database except as supported by ESRI. For example, Personal geodatabases are usually encountered only in the form of Access .MDB files, and not all DBMS systems are supported for use with SDE. There is a table storing the definitions of all coordinate systems, and a table storing the bindings between geometry columns and coordinate systems. Importing or linking an ESRI drawing into Manifold reads the binding and retrieves the coordinate system. Both Personal and SDE geodatabases store coordinate systems using a format similar to the PRJ extension introduced by ESRI to support storage of projected data within shapefiles. As a result of years of experience
Manifold® System Release 8.00 User Manual

encountering many different variations of PRJ files Manifold is able to read virtually all coordinate systems encountered in either Personal or SDE geodatabases.

**Manifold drawings** are the simplest of all in that when Manifold exports a geometry type and creates a spatial index it also creates a table named `MFD_META` to host metadata used by Manifold. All metadata is stored by Manifold in the `MFD_META` table, including information about coordinate systems (projections) used. Exporting a Manifold drawing writes the coordinate system into `MFD_META`. Importing or linking a Manifold drawing reads the coordinate system from `MFD_META`.

In general, Manifold always writes the coordinate system used during export of a drawing and always reads the coordinate system in use when importing or linking.

**Projection Matching During Export**

Different spatial DBMS vendors support different collections of coordinate systems (projections), as does Manifold. When exporting a drawing to a spatial database, Manifold matches the coordinate system of the drawing to the closest coordinate system supported by the database. If the database does not support the exact coordinate system of the component, Manifold will re-project objects on the fly.

To know the coordinate system that will be assigned to the exported component, press the [...] browse button near the projection readout in the export dialog. Manifold will show a dialog with a list of coordinate systems supported by the database with the intended coordinate system selected.

Sometimes DBMS vendors will use different names for what are the same coordinate systems, or may not even use obvious names. For example, each coordinate system in a PostgreSQL database is assigned a unique identifier, called an **SRID** (Spatial Reference system ID).

When exporting a drawing to a PostgreSQL database, pressing the [...] browse button near the projection readout in the export dialog will show a list of supported SRIDs together with the selected SRID. For example, for **Latitude / Longitude** the SRID value is **4030**.

In another example exporting drawings to SQL Server 2008 spatial uses EPSG codes to represent coordinate systems, so the value reported will be the EPSG code for the coordinate system in use.

Manifold will do its best to come up with a reasonable name for a coordinate system, if possible. For example, loading a coordinate system from a PostgreSQL data source parses the name of the coordinate system from its definition within PostgreSQL. Unnamed coordinate systems used in databases will be assigned names synthesized from their IDs (SRIDs).

**Units of Measure in Linked or Imported Surfaces**

A limitation of Manifold's mechanism for storing surfaces in spatial DBMS is that the linked or imported surfaces are expected to represent heights in **meters**. If a linked or imported surface is generated using a data set in which heights are expressed in **feet** or some other unit, then the hill shading effect will be overly dark. The workaround to this is to convert surface heights to meters before export to a database if they are in feet, using the Surface Transform calculator or an intermediate query. See the Queries and Images or Surfaces topic for an example query that will do the trick.

**Notes**

The idea of storing geometry within non-native geometry types in databases applies to Manifold itself, since, after all, tables within Manifold are in fact a dedicated database system of sorts. That's why Manifold can store geometry in the form of **GEOMETRY**, a binary data type, within Manifold's own tables. It is also why Manifold has binary types for other commonly-encountered GIS binary blob storage types, including **GeomWKB** for OGC WKB, **GeomSHP** for ESRI shapefile geodatabases and **GeomSDE** for SDE binary data. The availability of all these types makes it easy to convert geometry data between binary type formats used by different systems.

**GeomSHP** and **GeomSDE** are somewhat similar approaches with differences in binary data organization. **GeomSHP** stores coordinates as double-precision floating point values. **GeomSDE** stores coordinates as integer values which are compressed using a simple run-length encoding scheme. The coordinate values must be scaled and shifted using the information stored in metadata tables.
Note that there are no vendor-specific, spatial DBMS geometry types like Oracle's SDO_GEOMETRY within Manifold tables because those types are constructs that occur within the spatial DBMS. Manifold GEOMETRY (Geom) data is automatically converted back and forth with the target spatial DBMS geometry data type on data exchange with the spatial DBMS. In contrast, the various Geom types supported within Manifold tables are not specific to any one DBMS but can exist wherever binary types are allowed.

Manifold allows simultaneous usage of multiple geometry types within the same DBMS. For example, an Oracle database could contain drawings stored using Oracle's own SDO_GEOMETRY type using Oracle's own spatial indices, and that same database could also contain drawings stored using Manifold's Geometry type using generic spatial indices created by Manifold. In theory, it could also store an SDE geodatabase using ESRI-style technology. When Manifold connects to that Oracle database it will correctly identify all technologies in use and utilize appropriate methods when importing the Oracle, Manifold or ESRI types from that database.

It may seem odd to want to connect to a spatial DBMS and store spatial data within that DBMS while not using the native geometry type provided by that DBMS. There are cases where that nonetheless makes sense. For example, we may have the budget for only one enterprise-class DBMS server installation within our organization, and we may have chosen Oracle for that DBMS with the idea of standardizing upon Locator with SDO_GEOMETRY (Oracle's native geometry type) for our spatial geometry needs. But we might have occasion to want to provide data for interoperability with some client software that does not understand SDO_GEOMETRY but which understands OGC WKB. In that case, we could connect to our Oracle server and choose the Manifold type of spatial storage using Geometry (WKB) to enable export to the DBMS of drawings that are stored using OGC WKB.

In fact, quite remarkably we can link drawings to Oracle using different connection technologies for different drawings. One drawing could be stored using Oracle's native geometry. Another drawing could be stored using Geometry (WKB) managed by Manifold. We can move objects into the OGC form by simply copying and pasting from one drawing to the other.

Technical Support

DBMS packages used for spatial DBMS work are highly complex software systems with many installation, configuration, operation and administration options. The tips above presume reasonably default configurations set up by administrators who are reasonably worldly about configuring DBMS installations in a portable and open way. Successful operation with the various spatial DBMS packages Manifold understands is not normally difficult, but can require significant DBMS and IT technical expertise from database administrators when DBMS installations become more elaborate. For example, permissions can be set in DBMS installations that may conflict with use of clients like Manifold. This will come as no surprise to the DBA with client integration experience.

Manifold does not support third party DBMS products, with the exception that current Express editions of SQL Server (2008 and 2005), Oracle and DB2 will be supported for Enterprise Edition use with those DBMS products in default configurations. See comments in the Database Installations topic. Although the manifold.net team is enthusiastic about working with other DBMS products, such as PostgreSQL, the great variability encountered in open source installations does not make it economically feasible to provide technical support services beyond the three major commercial DBMS products.

Developer level support incidents may be used for questions about Enterprise Edition features used with the three supported database products, but only for the actual Enterprise dialogs. For example a question about connecting with Enterprise Edition via Database Console to an Oracle data source will result in an answer limited to the use of a specific Manifold dialog, such as explaining the purpose of the Server, User Name and Password boxes.

Support does not extend to explaining how you can configure users in the DBMS, how to determine what server name is being used or should be used or how to determine whether a given user has connectivity to a given data source. Use a database consultant to assist with DBMS-centric issues. There are many ways a clever (or overly clever or not so clever) database administrator can configure such complex DBMS products to make it impossible for clients like Manifold to work with them with the full feature set of either client or server - that's what keeps database integration consultants in business!

See Also

Spatial DBMS - The key introductory topic to spatial DBMS.

Data Storage Strategies

Database Installations
Manifold® System Release 8.00 User Manual

Database Administrator Edition
ESRI Geodatabases
Example: Storing a Drawing in Manifold Spatial DBMS
Example: Storing an Image in Manifold Spatial DBMS
Example: Storing a Surface in Manifold Spatial DBMS
Geometry in Tables
Linked Drawings
Manifold Spatial DBMS Facilities
Manifold Spatial Extender for SQL Server
Oracle Spatial Facilities
Project Pane - Open Data Source
Queries and Geoms
Spatial Extensions
SQL Server Spatial DBMS Facilities
Tools - Administrator Console
Tools - Database Console

Oracle Spatial DBMS Facilities
Oracle Spatial Facilities
Manifold Enterprise Edition provides built-in, highly-integrated facilities for working with Oracle databases using spatial DBMS technology. Manifold Database Administrator Edition also provides administrative capabilities used with Oracle database to enable and configure various Manifold features that may later be used from Enterprise Edition.

In particular, Manifold Enterprise Edition includes full support for use of Oracle's outstanding spatial technology, using either full Oracle Spatial licenses or simply via the Oracle Locator technology built into standard Oracle database products or, when available, Oracle's GeoRaster technology for storing images and other raster data. This topic provides an overview with links to other topics that discuss Oracle capabilities in detail.

Manifold is a complete, drop-in replacement for old-fashioned "middleware" as sold by legacy GIS vendors for Oracle Spatial applications. In many cases Manifold may be used as the user front end for Oracle Spatial applications or as the user interface, editing and visualization front end for geospatial data within standard Oracle databases (using Locator) without need to acquire and deploy Oracle Spatial licenses. Oracle's provision of core Oracle Spatial technology, such as the SDO_GEOMETRY type, within standard Oracle database versions makes it possible for enterprises to use Manifold to standardize on Oracle for all geospatial data needs in a highly flexible and efficient way.

Manifold support for Oracle spatial technology works throughout all of Manifold System, including the Manifold Internet Map Server and runtime license applications. Oracle spatial technology provides enterprises with an outstanding solution for geospatial data storage and exploitation. Manifold provides enterprises with a powerful and modern GIS front end to maximize their investment in Oracle through desktop GIS, Internet map serving or integration into custom applications.

Note that some Oracle products, such as Oracle Express Edition, may have Locator spatial capability but not provide GeoRaster capability. In that case Manifold can work with such Oracle products to save drawings but not images, which require GeoRaster capability.

Manifold Enterprise support for Oracle includes:

- **Oracle Call Interface (OCI) Integration** - OCI is the native Oracle interface for exchanging data. Using the native interface provides better performance and allows using features not exposed through generic database interfaces such as ODBC, OLE DB and ADO .NET. Use OCI when connecting to Oracle databases.
- **OCI Support for Tables** - Manifold can export, import and link table data read/write through OCI. SDO_GEOMETRY values are represented as Manifold native geometry values in the appropriate coordinate system. Manifold coordinate systems are automatically mapped to Oracle coordinate systems.
- **OCI Support for Drawings** - Manifold can export, import and link vector data (drawings) through OCI. Linked vector data sets (linked drawings) are read/write and are cached locally. This provides a superb user interface for working with spatial data within Oracle.

- **Dynamic Editing of Linked Drawings** - Manifold can dynamically display drawings created from SDO_GEOMETRY data within Oracle and allows use of the full range of interactive and programmatics Manifold tools, such as visual editing tools, transforms, adding/deleting/editing objects or table data and other Manifold features.

- **Multituser Drawing Editing with Version / Conflict Resolution** - Many users at the same time can edit the same drawing stored in Oracle. Manifold will automatically track the version of each object in case two users attempt to simultaneously edit the same object and will assist users in resolving any conflicts via the Review pane.

- **Import or Linking Drawings from Views** - Manifold drawings may be imported from or linked to Oracle views. Views are a very powerful way of extracting a portion of data using a query. See the Example: Storing a Drawing in Oracle topic for an example.

- **Area of Interest Windowing** - Users may import or link in only that part of an Oracle drawing or view that is within their area of interest. This allows storage of immensely large drawings in Oracle yet allows practical usage of a portion of such drawings. *Note:* Import or linking a drawing using an area of interest from a view assumes the view has a spatial index.

- **OCI Support for Rasters / Images** - Manifold can import, link and export raster data (images) with Oracle products that include GeoRaster technology. Linked raster data sets (linked images) are fetched from the database on the fly preserving coordinate information. Manifold can work with a variety of images stored in Oracle, including interleaved channel images, indexed (palette) images and other types. Manifold can import as a surface a single channel of an image stored in Oracle.

- **Export and Import of Surfaces** - Manifold can export surfaces to Oracle as GeoRasters and can also import surfaces stored as Oracle GeoRasters (however, Manifold releases up to and including 7.00 cannot link surfaces from Oracle GeoRasters). Export to Oracle and import from Oracle of images is similar to the export and import of images. See the Example: Storing an Image in Oracle topic for the process used both for images and surfaces.

- **Automatic Read / Write of SDO_GEOMETRY** - SDO_GEOMETRY is the fundamental data type used within Oracle spatial technology to store drawing objects. When exchanging data with Oracle data sources via OCI, Manifold automatically maps Manifold geometry columns into Oracle SDO_GEOMETRY data and vice versa.

- **Drawing Export to Oracle Databases** - Manifold can export drawings directly into either Oracle Spatial databases or into regular Oracle databases using Locator. Tables will automatically be created using SDO_GEOMETRY data for drawing objects and other data types for attributes. Manifold will automatically preserve projection information and will create a spatial index for the drawing. This facility is a great way of loading data into Oracle spatial databases.

- **Oracle Projections Support** - Importing or linking a raster or vector data set from an Oracle database brings the coordinate system of the data set into Manifold, which supports all Oracle coordinate systems. Exporting a vector data set from Manifold to an Oracle database lets the user select the coordinate system to use from the list of coordinate systems supported by the database. By default, Manifold selects the coordinate system available with Oracle which is the best match for the coordinate system of the data set, making any required coordinate system translation on the fly. Manifold recognizes Oracle coordinate systems defined through parameter tables as well.

- **Custom Database Console Support** - When connecting to an Oracle data source through OCI, the Database Console provides a variety of special Oracle features, such as using drawing icons to show tables that contain drawing geometry data, not available with other data sources. The Open Data Source command provides a rapid way to open the Database Console for components that are linked from an Oracle data source. *Note:* because the Database Console provides a more convenient user interface for browsing Oracle databases, we will almost always use the Database Console to import or link components from an Oracle database instead of using the File - Import or File - Link menu commands.

- **Database Administrator Edition** - Database Administrator Edition includes a powerful managerial console called the Administrator Console that enables IT and DBMS administrators to configure data sources used to store the enterprise's geospatial data. The Administrator Console allows administrators to configure GIS data storage within the data source so that the data source is easier and more convenient to use for ordinary GIS users. Database Administrator Edition also includes the Tools - Batch Export tool that facilitates rapid, mass uploading of Manifold drawings, images and surfaces into Oracle spatial databases.

The above dedicated capabilities are just the beginning of Manifold support for Oracle. Oracle's spatial technology provides enterprise-class geospatial database capabilities with unprecedented sophistication and power. Future editions of Manifold will extend and deepen Manifold's commitment to Oracle support, adding new capabilities at a rapid pace.
Manifold supports all Oracle projections (coordinate systems) although Oracle does not support all Manifold projections. When a component is imported or linked from an Oracle database it will automatically use that same projection within Manifold.

When a component is exported from Manifold to Oracle, Manifold will automatically choose the Oracle projection that is the best match, that is, the most similar to, the Manifold projection in use. Manifold export dialogs will report the degree of similarity between the Manifold projection and the proposed Oracle coordinate system. If need be, Manifold will re-project the component on the fly into that Oracle projection system as part of the uploading process.

There are two nuances of interest in the case of components using a coordinate system that appears both in Oracle and in Manifold (that is, all Oracle coordinate systems):

- Although Manifold supports all coordinate systems used by Oracle, sometimes the names used for those coordinate systems are different. For example, the Latitude / Longitude projection used in Manifold is known as the TWD97 projection in Oracle when using the WGS 84 datum (the default for Latitude / Longitude).
- Manifold defines coordinate system presets using higher-precision parameters than those used in Oracle so even if the coordinate system is otherwise identical the Manifold settings have to be lowered in precision to match those used by Oracle. Manifold understands such precision-matching requirements when identifying the equivalent projection in Oracle and will automatically re-project data to account for precision differences. If this or any other re-projection is not desired, Manifold provides a Do not project data option (off by default) that may be checked on to prevent any re-projection of data.

It should be emphasized that the second point above is an example of “overkill” accuracy on Manifold’s part and in no way is any practical limitation of Oracle.

**Frequently Asked Questions**

**Must I install something extra to get Oracle spatial capability in Manifold?** - No, not if you have Enterprise Edition installed (including Universal Edition and Universal Runtime). If you do not have Enterprise Edition or higher installed you will not be able to use Oracle spatial capabilities from within Manifold System.

**Must I use Database Administrator Edition to get Oracle spatial capability in Manifold?** - Strictly speaking, no, although it will be a very rare organization that will not acquire a few Database Administrator Edition licenses to support Manifold use with Oracle. Database Administrator Edition is required to “turn on” certain Manifold features such as friendly names and storage of drawing formatting within the database. Database Administrator Edition also provides conveniences such as batch upload of components and presets for imports and links that make life easier both for administrators and for ordinary users. If you are willing to do without the features and conveniences enabled for all users (including those using Enterprise Edition) by Database Administrator Edition then you could use Enterprise Edition only.

**What versions of Oracle can I use? Must I use Oracle Spatial?** - Manifold works with Oracle Spatial as well as other Oracle editions that include Locator (such as Oracle Express Edition) or GeoRaster (such as Oracle Enterprise Edition) capability.

**Must I install a Manifold license on my Oracle server machine?** No. Manifold is a client application that connects directly to the Oracle database server without needing any "middleware" or "server" piece of Manifold on that Oracle server. For example, you could have Oracle running on one or more server machines and then have Manifold installed on a few hundred client machines where operators are doing GIS. You need to install a Manifold license on the client machines, but you do not need to install Manifold licenses on the Oracle server machines. To take another example, suppose you had a web server hosting a web site that involved connections to an Oracle database on a different machine. If you had a GIS-enabled web site running on that web server machine using Manifold IMS you would need a Manifold license for that web server machine so that Manifold IMS could run. But, you would not need a Manifold license for the Oracle server machine.

**Is Manifold’s use of Oracle Spatial compatible with other applications?** Yes, if those other applications work directly with Oracle using standard Oracle interfaces and technologies. For example, if you load an Oracle Spatial database with vector data by exporting a drawing from Manifold any other application that knows how to work with Oracle Spatial databases will be able to use that vector data.

**Manifold runs in Windows - can I run my Oracle server on Linux?** Of course. Manifold is happy to connect to your Oracle data sources regardless of what operating system they use.
Does Manifold require ArcSDE to be able to connect to Oracle Spatial? No. Manifold requires no middleware or other software of any kind to be able to connect to Oracle databases. Manifold connects directly to Oracle data sources using the Oracle Call Interface (OCI).

Can Manifold connect to Oracle Spatial through ArcSDE? No, and there is no need to do so. Manifold works directly with Oracle Spatial. ESRI's ArcSDE is middleware designed to keep ESRI customers locked into ESRI products whether they like it or not. Manifold connects directly to Oracle and neither requires nor uses ArcSDE. Some customers who are stuck with ArcSDE applications for legacy reasons will no doubt continue to use ArcSDE; however, they can also use applications like Manifold to connect directly to the centralized Oracle Spatial database in addition to whatever they are doing via ArcSDE.

So Manifold is "DBMS-centric" rather than "GIS-centric?" For enterprises, yes. Clearly, if you have vast amounts of corporate data held within a supremely capable DBMS like Oracle your enterprise is DBMS-centric and darn well should be. It's madness to allow any GIS vendor to attempt to interpose itself as some sort of proprietary standard that interferes or limits in any way your ability to leverage your Oracle installations. While it's certainly true that some corporate processes involve such a high degree of specifically GIS functionality (such as heads-up digitizing of new parcels in a parcel map) that those particular processes are clearly GIS-centric, it is important to keep in mind that such processes are adjuncts to the core DBMS. They are simply examples of the rich extensibility and connectivity that a well-founded DBMS-centric geospatial data storage architecture allows. Manifold is committed to using native Oracle standards and technology as the guarantee of broad compatibility with many other applications and products. If Oracle standards are the touchstone, you can confidently deploy tools like Manifold knowing your core investment into Oracle will be protected.

Do Manifold queries execute locally or on the Oracle server? Queries written in the Database Console execute on the Oracle server. Other Manifold queries in your Manifold project will execute locally within Manifold spatial SQL. This means that if they involve data brought in from an Oracle database (such as if they involve a linked table) there will be a performance hit because data must be brought in from the Oracle server to allow the query to operate. We anticipate that future Manifold editions will extend the capabilities of Manifold's query processor so that queries will automatically be analyzed and those parts of the query that can be dispatched to run in the Oracle server will be run on the server.

See Also

Installing Oracle
Example: Storing a Drawing in Oracle
Example: Storing an Image in Oracle
Linked Drawings
Linked Images
Importing and Linking Tables
Geometry in Tables
Importing Drawings
Importing Images
Linked Images from Oracle Servers
Importing Tables
Linking Tables
Linking Drawings
Linking Images
Exporting Drawings
Export Drawing - Oracle
Export Image - Oracle
Exporting Tables
Map Server Overview
Tools - Database Console

Acknowledgements

The manifold.net team would like to thank Oracle for its support of independent software developers working to integrate products with Oracle. The Oracle Technology Network is a model of how a technology leader like Oracle encourages developers to undertake large engineering projects with confidence. This provides better integration and better value for Oracle users worldwide. Special thanks go out to the Oracle Spatial team in New Hampshire as well as to Oracle staff in California.
Installing Oracle

Manifold users will often use Oracle database products in conjunction with Manifold System. In particular, Manifold Enterprise Edition users will take advantage of Oracle Spatial Facilities that are provided within Enterprise Edition to store drawings and images within Oracle servers.

This topic provides a summary of installation procedures that may be used to install Oracle products for use with Manifold. Two products are covered: Oracle 10g Express Edition and Oracle 10g Enterprise Edition. In addition, Oracle Client software should be installed on machines that will be working with Oracle databases. The client software need not be installed on the machine on which the Oracle Express or Enterprise server software is installed. These software packages may be downloaded at no charge from Oracle’s web site and used to develop applications.

The following summaries have been written for installation on a Windows XP SP2 machine using Oracle versions that were downloadable at the time of this writing. Procedures for Windows Server 2003 and for newer Oracle editions will be very similar.

To install Oracle 10g Express Edition (10.2.1.0):

1. Close all running applications.
2. Start the installation of Oracle 10g Express Edition by launching OracleXE.exe in the Oracle installation files folder.
3. In the welcome screen click Next.
4. In the license screen read through the license and then click I accept the terms in the license agreement, and click Next.
5. In the destination location screen accept the default folder suggested for installation or specify a different folder if desired and click Next.
6. In the password screen enter a password string in the Enter password text box. Enter the same string in the Confirm password text box, and click Next. Write down the password you used.
7. In the summary screen note the port numbers used (For example, Oracle database listener: 1521, Oracle Services for Microsoft Transaction Server: 2030, HTTP listener: 8080), then click Install.
8. Wait until the installation completes.
9. In the installation complete screen click Finish. Close the database portal page, if you have any open.

Configuring Windows Firewall (This continuation of the installation is not necessary if you are not planning to access this instance of Oracle from other machines or if the firewall is turned off.)

10. Launch the Windows Control Panel.
11. Double click the Windows Firewall icon.
12. Switch to the Exceptions page. Click Add Port. Set (that is, enter) the name to Oracle Listener. Set the port number to the value noted in step 9 above (for example, 1521). Click TCP. Click Change Scope. Click My network (subnet) only. Click OK to close the Change Scope dialog. Click OK to close the Add Port dialog.
13. Click OK to close the Windows Firewall dialog.

After following the above steps it is easy to test that an Oracle Express installation is functional by using the Database Console in Manifold to quickly connect to Oracle. The following set of instructions assumes we are running Manifold on the same machine on which Oracle has been installed.

Connecting to an Oracle Express Installation:

1. Launch Manifold and launch the Tools - Database Console command.
2. Press the [...] browse button to the right of the Data source box to launch the Data Sources dialog.
3. Create a connection to Oracle Express and connect to it as given in the example in the Data Source Dialog topic.

Once a connection has been made to Oracle Express with Database Console as above the Data source string will remember that connection. Administrator Console (if we have Database Administrator Edition installed) will also remember that connection for the dialog's Data source.
To install Oracle 10g Enterprise Edition (10.2.0.1):

1. Close all running applications.
2. Start the installation of Oracle 10g Enterprise Edition by launching SETUP.EXE in the Oracle installation files folder.
3. In the installation method screen select Basic Installation. Set installation type to Enterprise Edition. Note the database name (orcl). Enter a password string in the Enter password text box. Enter the same string in the Confirm password text box, and click Next.
4. In the prerequisite checks screen click Next.
5. In the summary screen, click Install.
6. Wait until the installation completes.
7. In the database configuration assistant screen that pops up during the installation, click OK.
8. If you get a Windows Security Alert box asking whether or not you want to block java or javaw, click Keep Blocking.
9. At the end of the installation screen click Exit. In the confirmation dialog click Yes.

Configuring Windows Firewall (This continuation of the installation is not necessary if you are not planning to access this instance of Oracle from other machines or if the firewall is turned off.)

10. Launch the Windows Control Panel.
11. Double click the Windows Firewall icon.
12. Switch to the Exceptions page. Click Add Port. Set (that is, enter) the name to Oracle Listener. Set the port number to the value noted in step 9 above (for example, 1521). Click TCP. Click Change Scope. Click My network (subnet) only. Click OK to close the Change Scope dialog. Click OK to close the Add Port dialog.
13. Click OK to close the Windows Firewall dialog.

To install the full version of Oracle client software and configure it connect to an instance of Oracle (not required if Manifold is used on the server machine):

1. Close all running applications.
2. Start the installation of Oracle 10g Client by launching SETUP.EXE in the Oracle installation files folder.
3. In the welcome screen click Next.
4. In the installation type screen select Custom and click Next.
5. In the home details screen specify the installation folder or accept the default and click Next.
6. In the product components screen check Oracle Call Interface, Oracle Net as well as any other components you want to install and then click Next.
7. In the prerequisite checks screen click Next.
8. In the summary screen click Install.
9. Wait until the installation completes.
10. In the welcome screen of the Oracle Net Configuration Assistant that comes up at the end of the installation process, be sure the Perform typical configuration box is unchecked and click Next.
11. In the naming methods screen of the assistant click Next.
12. In the service name screen of the assistant specify the service name as the database name you used when installing an instance of Oracle (by default, xe for Express Edition and orcl for Enterprise Edition), and click Next.
13. In the protocols screen of the assistant select TCP and click Next.
14. In the TCP / IP configuration screen of the assistant specify the host name as the name of the machine running Oracle and click Next.
15. In the test screen of the assistant click Yes, perform a test and click Next. If the test does not succeed (if it fails the text output will contains the phrase "The test did not succeed."), click Change Login and supply a valid user name (system) and the password you used when installing the instance of Oracle and click OK. After the test succeeds click Next.
16. In the net service name screen of the assistant specify the net service name as something memorable, such as oracledb and click Next. You will use this name when connecting to this database from Manifold and other applications.
17. In the another net service screen of the assistant click No and then click Next.
18. In the done screen of the assistant click Next.
19. In the second done screen of the assistant click Next.
20. In the final done screen of the assistant click Finish.
21. In the end of installation screen click Exit. In the confirmation dialog click Yes.

To install Oracle Instant Client and configure it to connect to an instance of Oracle (not required if Manifold is used on the server machine):

(This process is also described in the readme file in the archive of installation files downloaded for Oracle client software.)

1. Create a new folder (for example, C:\Oracle) and unzip the archive into this folder.
2. Set the system PATH environment variable to include the path to the folder.
3. Set the system TNS_ADMIN environment variable to point to the folder.
4. Create a new blank text file and enter the following text:

   localservice=
   (DESCRIPTION=
     (ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=server)(PORT=1521)))
     (CONNECT_DATA=(SERVICE_NAME=remoteservice)))

   Replace localservice with the name to use for the database connection on the machine to which you are installing Instant Client, server with the name of the machine hosting the instance of Oracle you will use, and remoteservice with the name of the database to connect to on that machine (for example, xe if it is the default database name for Oracle Express Edition or orcl if it is the default database name for Oracle Enterprise Edition).
5. Save the text file as TNSNAMES.ORA in the folder created in step 1.

To connect to an Oracle database using Manifold:

1. Set up a connection to the desired database using either the full version of Oracle client software or Oracle Instant Client. This step is not required if Manifold is used on the server machine.
2. Launch Manifold.
3. Launch Tools - Database Console.
4. Click the Browse button to connect to the Data Source dialog.
5. In the Data Source dialog create a connection to the desired Oracle database and connect to it following the example given in the Data Source Dialog topic. When specifying the server name in the in the Oracle Data Source dialog that appears as part of specifying an Oracle data source in the Data Source dialog, specify as the server as:
   a) The Windows computer name of the machine being used if Manifold is being used on the server machine. For example, if the Windows Control Panel, System applet's Computer Name tab says the name of our machine is gisfiles that is what we would use.
   b) If Manifold is being used on a client machine that is different from the server machine, the name of the database connection as configured in the Oracle client in use. For example, oracledb if we follow the above example for installing the full version of Oracle client software or localservice if we follow the above example of installing Oracle Instant Client and haven't bothered to change the connection name from "localservice". Note: the default server name used for Oracle Express installations is xe and is not oracledb. This is an important detail that is different between Oracle Express and Oracle Enterprise editions.

Note that in the above procedures, although there are the expected dialogs to go through for installing Oracle software as there would be for any big-time application with sophisticated options, once the installation is done it is very easy to connect to Oracle from Manifold. Manifold will remember the data source used in both the Database Console and (if Database Administrator Edition is installed) the Administrator Console.

Adding Users to Oracle
For security and administrative control, it is a good idea to create user logins for an Oracle database so that users can access the database without having to connect using Administrator logins such as `system`.

For Oracle XE and other Oracle products, Oracle provides a convenient Database Control facility running over local web pages that may be used to add users. Launch the Database Control from the Windows Start button, drilling down through All Programs to the Oracle program choices. After logging into the Database Control using the `system` administrator login we can go to the Administration tab and drill down into the Users page. Click the Create button to launch the Create User page.

In the General tab, provide a Name and Password for the user and choose a Default Tablespace (such as USERS) and a Temporary Tablespace (such as TEMP) for the user.

In the Roles tab, the user must have a CONNECT role, which is provided by default.

In the System Privileges tab, the user must have CREATE SEQUENCE, CREATE TABLE and CREATE TRIGGER privileges. Click the Edit List button and add these privileges.

In the Quotas tab, the user must be allowed to use sufficient space for their storage requirements. Many DBMS administrators in small organizations will give users Unlimited quotas for USERS and TEMP tablespaces. This is somewhat trusting of users, but in an era of very large hard disks and reasonably sensible users it often is easier to simply grant unlimited usage than it is for relatively inexpert DBMS administrators to worry about managing storage space quotas.

Press OK to create the user. We can then use that user login to connect to the DBMS. Each login that uses the USERS tablespace will be able to see drawings or images stored in the DBMS by other such logins.

DBMS administrators will find that Oracle has many first-rate facilities for organizing databases and user access. Such facilities are beyond the scope of this documentation but are well-covered by Oracle documentation and educational resources.

See Also

Oracle Spatial Facilities
The Data Source Dialog
Tools - Administrator Console
Example: Storing a Drawing in Oracle

In this example we export a drawing to an Oracle database and then we link it into a Manifold project. This example requires Manifold Enterprise Edition for Oracle Spatial capability and also requires an Oracle server that includes Locator capability, such as Oracle Spatial, Oracle Express Edition or Oracle Enterprise Edition.

In addition to showing the mechanics of uploading a drawing into Oracle Spatial and linking a drawing from Oracle Spatial, this example also shows off Manifold's ability to dynamically, visually edit drawings that are stored within Oracle Spatial, even within a multi-user editing environment.

Although of course Manifold also supports use of SQL to edit and otherwise manipulate drawings within Oracle Spatial, it's nice to know that if we prefer visual, interactive, mouse and command based editing to work with our Oracle Spatial data we can apply the full power of interactive Manifold editing just as if the drawing was a local part of our Manifold project.

Step 1: Export the Drawing to Oracle

We will use the example drawing of province boundaries in Mexico provided on the Manifold downloads site and used in various topics in the Examples chapter such as the Turning Layers Off/On by Zoom topic.

Our project contains just this single drawing. To upload it into an Oracle server we either right click on the drawing in the project pane and choose Export, or with the focus on the opened drawing window we choose File - Export - Drawing. In the resulting Export Drawing dialog we choose Databases () in the Save as type box. This launches the Data Source dialog.
If we have not yet created a connection to the Oracle server desired, we do so using the instructions in the example in the Data Source Dialog topic. When the data source is created we click on it to highlight it and press OK.
In the **Export Drawing** dialog we change the **Type** to **Oracle** to save the drawing using native Oracle Spatial geometry (as opposed to generic Manifold spatial storage using some other geometry type).

We will use default settings and export all of the checked columns with the drawing to Oracle. If we scrolled farther down into the **Columns** panel we would see that intrinsic columns are not checked by default. We could also note that neither the **ID** column nor the **Geom(I)** column appear because both are automatically exported to Oracle in all cases.

**Note:** Oracle supports only one variable-length column (such as binary data variable length or ANSI text variable length columns, for example) per table. If more than one of the columns to be exported is a variable-length column the exporter will complain and ask us to convert some of those columns into a different type (for example, we could convert them into fixed-length columns).

We press **OK** to upload the drawing. Uploading such a small drawing is fast. When the dialog finishes, we can see if there is indeed a new drawing exported into our Oracle server by using the Database Console to take a look at what is in the server.

**Step 2: Find a Drawing Stored in an Oracle DBMS**

We launch Tools - Database Console to launch the database console dialog.
The dialog remembers the last-used data source. We press the Refresh toolbar button to connect to the Oracle data source.

Whoa! What’s all that? When the database console connects to an Oracle server it shows us all the database objects in that server. If we connect to an Oracle Enterprise Edition server that has been installed with the full set of Oracle examples there will already be many objects containing geometry in that server. An actively used Oracle server can easily contain thousands of database objects.

However, lucky for us Manifold provides a Filter box that allows us to display only those objects of interest.
We can enter whatever characters we like in the Filter box to filter the view to only those objects that contain that exact string in their names. As we begin to enter characters, Manifold will begin searching and filtering the view (no need to press Enter - just type the filter string). Note that the Filter pattern is case-insensitive, so we can rapidly write what we like without needing to use the SHIFT key.

As we enter the Filter string, the view display reduces down to only the two database objects that contain the Filter string. Note that the top object has a Manifold drawing icon displayed, so we know it is something that can be imported or linked into a Manifold project as a drawing.

**Note:** In most Manifold installations using Oracle there will be one or more DBMS administrators who can use Database Administrator Edition to configure friendly names (also called component names) like those used for regular Manifold components in Manifold projections. If friendly names have been configured for a database then Database Console will automatically configure itself to use friendly names and we won’t have to wade through so many database objects that are not of interest.

However, even in that case there can be hundreds or even thousands of Manifold components stored in an Oracle database so the Filter box will still come in handy.

**Step 3: Link a Drawing Stored in an Oracle DBMS**

We can either import the drawing from the Oracle database, or we can link to it to create a dynamic, linked drawing in the project. We will link the drawing, as this keeps the drawing stored in the Oracle server with a dynamic representation of the drawing appearing in the local project.
To do so, we click on the drawing name to highlight it and then we press the **Link** button. We then press the **Close** button to close the database console.

When we do so, a new linked drawing appears in Manifold in the project pane. In this example, we linked the drawing right back into the same project from which we exported the drawing, but we could have just as easily closed Manifold and linked this linked drawing into a new project or into any project. The linking process is almost instantaneous.

**Note:** If Database Administrator Edition has been used to pre-configure import and link options for a component the component will link without any intermediate dialogs raised. If the component has not been pre-configured with options for us then a dialog with link options will be raised to allow us to specify any options desired.

**Step 4: Edit a Linked Drawing Stored in an Oracle DBMS**

We can click open that linked drawing and work with it as we might any other drawing. Let us do some editing using **smart mouse editing** moves as set forth in the Editing Objects topic, with which we assume the reader is familiar.
For example, we can **CTRL-ALT** click onto an object in the drawing to select it for editing. In the illustration above we **CTRL-ALT** click onto the province of Durango.

Selecting an object for editing in a Manifold drawing creates editing handles at the coordinates that define the geometry of the object. We can then drag those handles to re-shape the object, or do other cool things with them. For example, if we **SHIFT** click and drag a handle we move the entire object.
Let's **SHIFT** click and drag one of the editing handles on Durango and move the province to the Northeast out into the ocean.

When we release the **SHIFT** click and drag the province moves to the new location. We can then click anywhere in the drawing to deselect the object.
Like magic, there Durango is out in a new home as an island. The really amazing thing about this is that as we edit the drawing we are actually working with data stored not in the local Manifold project but brought into Manifold dynamically from the Oracle Server. Changes to the drawing are committed to the Oracle server as soon as they are accomplished.

**Step 5: Edit a Linked Drawing’s Table**

If we like, we can even make changes to the drawing’s table as well. Just as we can change the geometry of objects (that is, their shapes), we can change their data attributes as well.

![MEXICODRAWING Table](image)

<table>
<thead>
<tr>
<th>ID</th>
<th>ID 2</th>
<th>SQMI</th>
<th>PLACE_NAME</th>
<th>SQKM</th>
</tr>
</thead>
<tbody>
<tr>
<td>39</td>
<td>1</td>
<td>26635.63</td>
<td>SAN LUIS POTOSI</td>
<td>63068</td>
</tr>
<tr>
<td>40</td>
<td>2</td>
<td>32009.10</td>
<td>ZACATECAS</td>
<td>73252</td>
</tr>
<tr>
<td>41</td>
<td>3</td>
<td>52374.05</td>
<td>DURANGO</td>
<td>12318</td>
</tr>
<tr>
<td>42</td>
<td>4</td>
<td>76106.63</td>
<td>SONORA</td>
<td>182050</td>
</tr>
<tr>
<td>43</td>
<td>5</td>
<td>23603.34</td>
<td>SINALOA</td>
<td>58328</td>
</tr>
</tbody>
</table>

We click open the table and then double-click into the **PLACE_NAME** cell for Durango to change the name. [This illustration shows an example table where we have dragged this column leftwards in the table so the illustration can be smaller and still show the column.]

![MEXICODRAWING Table](image)

<table>
<thead>
<tr>
<th>ID</th>
<th>ID 2</th>
<th>SQMI</th>
<th>PLACE_NAME</th>
<th>SQKM</th>
</tr>
</thead>
<tbody>
<tr>
<td>39</td>
<td>1</td>
<td>26635.63</td>
<td>SAN LUIS POTOSI</td>
<td>63068</td>
</tr>
<tr>
<td>40</td>
<td>2</td>
<td>32009.10</td>
<td>ZACATECAS</td>
<td>73252</td>
</tr>
<tr>
<td>41</td>
<td>3</td>
<td>52374.05</td>
<td><strong>DURANGO NORTE</strong></td>
<td>12318</td>
</tr>
<tr>
<td>42</td>
<td>4</td>
<td>76106.63</td>
<td>SONORA</td>
<td>182050</td>
</tr>
<tr>
<td>43</td>
<td>5</td>
<td>23603.34</td>
<td>SINALOA</td>
<td>58328</td>
</tr>
</tbody>
</table>

Since we have dragged Durango more or less Northward in the drawing and further, since we know the Spanish for “North” but not for “Northeast” we will rename the province to **DURANGO NORTE**.
Press Enter and the edit to the table cell happens. To commit these changes to the server, we save the project.

**Step 6: Multiuser Viewing and Editing**

As mentioned before, the cool thing about all this is that we are editing Oracle data stored within our Oracle server, which could be accessed simultaneously by tens, hundreds or even millions of other people.

For example, we could open a second Manifold session on our machine, or someone else on some other machine that can access the same Oracle server could open a Manifold session on their machine and also link in the very same Mexico drawing we are using.

When they link in the drawing and open it (the illustration above shows a different Manifold session going on at the same time as the session in which the example was done), they will see that in this particular drawing of Mexico the province of Durango is indeed out to sea.

**Step 7: More Editing**

In either session of Manifold we can do even more dramatic editing than simply moving objects. We can add them or delete them or do whatever else we like.
We can click on Durango to select it and then press the **Delete** key.

It's gone! No more Durango Norte in the ocean. If we like, we can create a new object.
Using the Insert Area command, we can draw a triangular area free hand.

The new area appears in our drawing.

| 72 | 35 | 22205.42 | CAMPECHE | 50812 | 57 |
| 73 | 36 | 2445.07  | COLIMA   | 5191  | 42 |
| 74 | 37 | 2145.36  | MORELOS  | 4950  | 11 |
| 75 | 38 | 653.44   | DISTRITO FEDERAL | 1499 | 82 |
| 76 | 39 | [Isla Manifolds] |

We click open the drawing's table and see that the last row in the table (added when an object was added to the drawing) has no values. We'll add a name for our new island object. Changes made to the table will be committed to the server as soon as they are made.

Going back to our other session of Manifold we can choose View - Refresh Data to update any open windows with the latest changes. As expected, we see that Durango has been deleted and a new triangular area has been added to the drawing.
This example is deliberately made simple. However, in real life we could use linked drawings stored in Oracle databases for applications such as maintaining real estate parcel drawings for cities and towns. A city could centralize its geospatial storage within an Oracle database where many different applications and business processes could use that data.

As the data needs changing, such as when parcels are subdivided or changed, the data can be edited interactively using Manifold by those who have access permissions to do so. Those changes will be available to all Manifold sessions linking to that Oracle database. For example, a Manifold IMS web page could include a drawing layer linked from that Oracle database. Changes made to objects in that Oracle database will immediately be reflected in what that web page displays.

Note

The default name of the identity column used when exporting a drawing to an Oracle database is **OID** to avoid naming collisions with the **ID** column that stores Manifold object IDs in Manifold drawings.

If we have Database Administrator Edition we can use the Tools - Batch Export command to export many drawings at once to Oracle.

**Tech Tip: Deletions in Database Console**

It's easy to delete unwanted database tables by writing a drop table query in the query pane at the bottom of the Database Console.

```
drop table SYSTEM.US_MAIN
```

Simply enter `drop table` and then double-click on the table (drawings and other images are stored as tables in Oracle) in the upper pane to add that name to the query without need to manually enter it. Press the ![run button](image) to launch the query and the table will be deleted. Note that the upper list of objects in the database will not be refreshed to show the deletion until we press the ![refresh button](image).

Deletions of spatial components such as drawings are more complex than simply deleting a table. See the discussion in the Database Console topic.

**Tech Tip: Using Views**

One outstanding advantage of using Oracle to store drawings is that we can use the server-side power of the database to grab subsets of data by using Oracle Views. We can create a view by using a CREATE VIEW statement in Database Console and then link to that view.

A view is simply some SQL that grabs a desired subset of objects. For example, the following SQL will return all objects in `mydrawing` within a rectangle from (20, 30) to (120, 130):

```
--SQL
SELECT * FROM mydrawing WHERE SDO_FILTER(mydrawing.geom,
  SDO_GEOMETRY(2003, NULL, NULL,
  SDO_ELEM_INFO_ARRAY(1, 1003, 3),
  SDO_ORDINATE_ARRAY(20, 30, 120, 130))) = 'TRUE';
```

Using SQL statements in Database Console we can create a view in a way that makes it ready for use as a subset of a drawing in Manifold.
For example, suppose we have a latitude / longitude drawing called world that has been exported from Manifold. We would like to create a view called worldview that shows only those objects covered by a rectangle between latitude 20 and 30 and between longitude 120 and 130.

To be able to change and delete existing objects as well as edit new objects we must also specify the view has a primary key and also enter the view into the geometry metadata table.

To specify the view has a primary key we could use:

```sql
--SQL
CREATE OR REPLACE VIEW worldview(oid PRIMARY KEY DISABLE NOVALIDATE, version, geometry) AS
SELECT * FROM world WHERE SDO_FILTER(geometry, SDO_GEOMETRY(2003, 2000097, NULL,
  SDO_ELEM_INFO_ARRAY(1, 1003, 3), SDO_ORDINATE_ARRAY(20, 30, 120, 130))) = 'TRUE'
```

To enter the view into the geometry metadata table we could use:

```sql
--SQL
INSERT INTO USER_SDO_GEOM_METADATA VALUES('worldview', 'geometry',
  MDSYS.SDO_DIM_ARRAY(
    MDSYS.SDO_DIM_ELEMENT('Longitude', -180, 180, 1),
    MDSYS.SDO_DIM_ELEMENT('Latitude', -90, 90, 1)), 2000097)
```

Given just the above SQL, any newly added objects will not become editable until we do a View - Refresh Data command. To enable newly added objects to become immediately editable we have to create a synonym for the sequence that is used to produce unique keys for the drawing so that this sequence is available in the view as well:

```sql
--SQL
CREATE SYNONYM worldview_seq FOR world_seq
```

The above example illustrates a view containing objects covered by the specified latitude and longitude extents. We could, of course, write the SQL however we desire to select specified objects using whatever criteria we like. For example, suppose our drawing shows oil wells that have a date attribute that specifies when the well was drilled. We could create a view that showed only those wells drilled between desired dates.

**See Also**

Database Administrator Edition
Export Drawing - Oracle
Installing Oracle
Oracle Express Edition
Oracle Spatial Facilities
Tools - Database Console
Tools - Batch Export
Example: Storing an Image in Oracle

In this example we export an image to an Oracle database and then we link it into a Manifold project. This example requires Manifold Enterprise Edition for Oracle Spatial capability and also requires an Oracle server that includes GeoRaster capability, such as Oracle Spatial or Oracle Enterprise Edition. To export an image into an Oracle Spatial database we must have access to that database and have write permissions. The process set forth in this topic also works for exporting surfaces to Oracle GeoRasters as well.

Oracle Express Edition does not include GeoRaster capability so this example will not work with regular Oracle distributions such as Oracle Express. Oracle does make it possible for people to download for test and evaluation free copies of Oracle products that do include GeoRaster capability. See the Oracle web site for the latest information on such free downloads. See the Export Image - Oracle topic for notes on the Export Image dialog as used with Oracle. See the notes at the end of this topic for a Manifold solution to storing images within regular Oracle distributions.

This example assumes the reader is familiar with Manifold images. See the Images and subsequent topics for details on Manifold images.

**Step 1: Export the Image to Oracle**

We begin with the SanFran example image that may be imported from the SanFran.jpg sample file on the Manifold downloads site. This is an RGB image that is 1514 x 1440 pixels in size.

To export an image to an Oracle database:

1. Open the image in an image window.
2. Choose File - Export - Image from the main menu.
3. In the Export dialog choose Data Sources () in the Save as type box.
4. Connect to the desired Oracle data source, creating a data source for the Oracle server desired if one has not already been created in the Data Source dialog.
5. Provide the information required in the Export Image dialog. Choose Oracle in the Type box to export using Oracle GeoRaster if available. Choose Manifold in the Type box to use Manifold image storage within the database. Check the other options as desired.
6. Press OK.
In the illustration above we have connected to an Oracle server that was named Oracle Spatial Server in the Data Source dialog. We have chosen Oracle in the Type box to use Oracle’s native GeoRaster technology.

Note: If we want to manually choose settings for GeoRaster storage in Oracle we can click the [...] browse button to the right of the Name box to launch the Oracle Image Name dialog. See the technical note at the end of this topic for more information.

We use default settings for this dialog, accepting the recommended defaults such as checking the Create pyramids box. This creates additional views of the image at different scales and stores them into the Oracle database so that later on when we retrieve the imager and view it as we zoom in and out of the image there will be redundant views to make the zoom process look faster. It's a very good idea to create pyramids, especially for large images, so don't uncheck this default unless you like slow and unresponsive images. Press OK.

Uploading such a relatively small image is fast. When the dialog finishes, we can see if there is indeed a new image exported into our Oracle server by using the Database Console to take a look at what is in the server.

Note: Export of images to Oracle is robust. For example, exporting an image to Oracle as a tiled table correctly rolls back all changes upon a failure.

**Step 2: Using an Image Stored in an Oracle DBMS**

We launch Tools - Database Console to launch the dialog and then we press the Refresh toolbar button to connect to the Oracle server. The dialog remembers the last-used data source for us.
Our Oracle server happens to be loaded with hundreds of items, so we enter *sanfran* into the Filter box to show only those items that have the string "sanfran" (case insensitive) somewhere in their names. We see right away there is indeed an image in the Oracle server with the desired name.

We can see information on that image by clicking it to highlight it. The status bar at the bottom of the display shows that it is a 3 channel image (as expected for an RGB image) and is 1514 x 1440 pixels in size. We can bring it into our project by clicking on the Link button in the toolbar. We then press the Close button to close the database console.
When we do so, a new linked image appears in Manifold in the project pane. In this example, we linked the image right back into the same project from which we exported the image, but we could have just as easily closed Manifold and linked this linked image into a new project or into any project. The linking process is almost instantaneous.

If we click open the image linked in from Oracle we see it is exactly the same as the image we uploaded from Manifold. There may be one thing we notice that's different once we try this example out on really big images: images linked in from Oracle are **fast**... really fast! It's like using an ECW compressed image in that the combination of fast Oracle DBMS image storage, fast access and manipulation by Manifold and fast Manifold exploitation of pyramids results in images that are much faster to open, pan and zoom than ordinary local images.

Note that this image is a **linked** image so it is coming in on the fly from the Oracle server. It is not stored in the local Manifold project. If we have a fast connection to our Oracle server we will be pleased with the performance. If we link the image through some appallingly slow Internet connection to an Oracle server sitting on the other side of the planet we might not be so happy, but in that case the problem is the slow Internet connection and not the speed of the server.

If we wanted to **import** the image from the Oracle server we could have just as easily done that in the database console by highlighting the image and then clicking the **Import** button in the console’s toolbar. That would have imported the image into a local RGB image that would be stored within the Manifold project.

**Step 3: Import a Surface from an Image Channel**

Come to think of it, we can show an example of importing a raster something from an Oracle database by showing off Manifold's ability to bring in surfaces from image channels. This is a capability that is often used when "images" stored in Oracle are really raster data sets that consist of more than one channel where each channel encodes some value such as terrain elevation, temperature or other value that is represented as a surface.

We don't have such a data set for this example, but we do have an RGB image with 3 channels (Red, Green and Blue) already exported into our server. We can import one of the three channels, making pretend it is surface data. This is a little weird, but it shows the process. [If the image was really a stored surface, it would also have a Height channel that could be imported to reconstitute the surface.]
We open up the database console one more time and connect to the Oracle server by pressing the **Refresh** button on the toolbar. We once again enter *sanfran* into the **Filter** box so that instead of showing us hundreds of items in the database the console only shows us those items with "sanfran" in their name.

As before, the console shows us the *SanFran* image we uploaded. This time we click the + box next to the image name to expand that image’s hierarchy.

We can see that there are three channels under the image. If we click one of them to highlight it we can see that the status bar at the bottom of the dialog reports that it is 8-bit unsigned Integer data.

We click the bottom channel to highlight it and then press the **Import** button in the toolbar.
A new surface and matching terrain appear in the Manifold project pane. Note that this is not a linked surface because it was imported and not linked from the Oracle server.

If we click open the surface we see a shaded relief surface created from the Blue channel of the RGB image. The surface has been interpreted from the values of the pixels, treating their color value as if that color value represented a height. Blacks are "low" terrains with all black being a height of zero while whites are "high" terrain.

We can view the terrain to see this effect. The view is roughly to the north and shows the upper two thirds of the terrain as seen when hovering over the ocean Southwest of Monterey Bay. The broad, lighter swath of higher "terrain" in the foreground is Silicon Valley, which is light grey-green in the image.

The screenshot above was created by setting the Z scale in the Terrain - Surface dialog to a value of 0.10 to minimize the rather abrupt transition between pixels. A Black to white palette was used and 64 tiles are being displayed so that the surface can be seen in its entirety when hovering above it. No clouds or water are used, although there is some fog to color the sky.

**Step 4: Uploading a Palette Image**
Suppose we start with a palette image instead of with an RGB image?

For example, suppose we create a *SanFranPalette* image like the one above, which is the *SanFran* RGB image converted into a palette image using adaptive interpolation and sixteen colors.

We can upload this image into Oracle using exactly the same procedure as in Step 1 above.

However, when we look at our server using Database Console we can see that the new image consists of only one channel and not three.

**Note**

If we have Database Administrator Edition we can use the Tools - Batch Export command to export many images or surfaces at once to Oracle.

**Technical Note: Oracle Image Name Dialog**
Normally the Export Image dialog is used without manual configuration of Oracle GeoRaster storage. Experts who are deeply familiar with the inner workings of Oracle GeoRaster storage may want to manually adjust default settings.

If we want to manually choose settings for GeoRaster storage in Oracle we can click the [...] browse button to the right of the Name box to launch the Oracle Image Name dialog.

Controls

- **Georaster table**: Name of the raster reference table.
- **Georaster column**: Name of the column in the raster reference table.
- **Raster data table**: Name of the raster definition table.
- **Raster ID**: ID of the raster in the raster definition table (an integer value).

Raster Schemas

The names of Manifold images stored in Oracle can use one of the following schemas:

- RDT
RDT is the name of the raster definition table, RASTERID is the ID of the raster in the raster definition table (an integer value), RASTERTABLE is the name of the raster reference table, and RASTERCOLUMN is the name of the column in the raster reference table.

In addition to recognizing the above schemas, the export dialog also allows editing all parts of the image name explicitly using the Georaster table, Georaster column, Raster data table and Raster ID values.

To understand what these schemas mean, let's take a look at how Oracle stores images (that is, rasters). Oracle stores images in a relatively complex, but scalable, way.

Images are made up of tiles. The tiles for each particular image are stored in a table called a Raster Definition Table or RDT for short. An RDT can contain tiles for more than one image, with each image having a unique ID called a RASTERID above.

An Oracle database can have more than one RDT. There normally is at least one more table called a Raster Reference Table or RRT for short and called RASTERTABLE above, which lists the RDTs and the images they contain.

An Oracle database can have more than one RRT. Finally, an RRT can technically have more than one set of columns referring to RDTs, called RASTERCOLUMN above.

Normally, when uploading an image to Oracle we don't care about all this stuff and will just enter the name to be used for the exported image without any dots. For example, we might use the name SanFranciscoBayArea.

When that is done Manifold will use the first schema and will use the entered name as the name of an RDT. Manifold then will then assume the default RASTERID (a value of 1) and the default RASTERTABLE (whatever is the first found RRT) and the default RASTERCOLUMN (whatever is the first applicable set of columns).

If using the default values above identifies an existing raster already in the database, after popping open a confirmation dialog Manifold will overwrite it. If the values do not identify any existing raster, Manifold creates a new raster.

Notes

This example uses Oracle GeoRasters, as found in Oracle's flagship Oracle Spatial product. However, using standard Manifold spatial DBMS facilities we can save images at very high performance using regular Oracle distributions. For examples, see the Example: Storing an Image in SQL Server 2008 and Example: Storing an Image in Manifold Spatial DBMS topics.

See Also

Export Image - Oracle
Images
Image Types
Installing Oracle
Intermediate Levels and Pyramids
Tools - Batch Export
Oracle Spatial Facilities
SQL Server Spatial DBMS Facilities

Manifold Enterprise Edition provides several ways of performing spatial DBMS work with Microsoft SQL Server:

- **Microsoft SQL Server 2008** - This new Microsoft product provides spatial support using native spatial DBMS capability such as Oracle Spatial or IBM DB2 with Spatial Extender. Manifold Enterprise Edition and higher editions have built-in capability to work with native SQL Server 2008 spatial capabilities, including both GEOMETRY and GEOGRAPHY data types. Manifold has supported SQL Server 2008 spatial capabilities since the very first SQL Server 2008 spatial pre-release in 2007. See the Example: Configuring SQL Server 2008 topic before working with SQL Server 2008.

- **Microsoft SQL Server 2005 with Manifold Spatial Extender** - Manifold has created a spatial extender for SQL Server 2005 to enable users to work with SQL Server 2005 as a high-performance spatial DBMS until SQL Server 2008 is released. The Manifold spatial extender utilizes Manifold geometry and Manifold spatial indices with a small amount of code that is uploaded into the SQL Server 2005 server for execution server-side. The uploaded code enables SQL Server 2005 to achieve better performance than a purely generic use of spatial indices. The Manifold Spatial Extender also works with SQL Server 2005; however, if we have SQL Server 2008 we should use SQL Server 2008's spatial capabilities. **Note:** When used with SQL Server Express 2005 the Manifold Spatial Extender requires SQL Server Express 2005 SP2.

- **Microsoft SQL Server with Manifold Generic Spatial Index** - This applies to SQL Server 2000 and more recent SQL Server editions such as SQL Server 2005 and SQL Server 2008. Because SQL Server, like almost every other big-name DBMS, provides binary storage we can use Manifold's generic, non-native spatial DBMS capability within SQL Server using whatever Manifold-supported geometry type we like (such as Manifold's Geometry or OGC WKB) together with Manifold-created spatial indices. This usage of SQL Server is just as we would use any other ordinary DBMS, such as MySQL, as a spatial DBMS through Manifold-managed spatial facilities. For performance reasons, with SQL Server 2005 it is always a better idea to use the Manifold Spatial Extender for SQL Server than to use generic spatial indices. Manifold generic spatial indices also work with SQL Server 2008; however, if we have SQL Server 2008 we should use SQL Server 2008's spatial capabilities.

- **Microsoft SQL Server 2005 with ESRI SDE** - ESRI products can utilize a SQL Server 2005 installation to host an ESRI SDE geodatabase. If SQL Server 2005 hosts an ESRI SDE geodatabase Manifold can connect to that ESRI SDE geodatabase and import and link drawings using ESRI conventions. It is possible for SQL Server 2005 to be used by ESRI products to host Personal geodatabases as well, in which case Manifold can also connect and work with that geodatabase like the SDE case. However, as of the present writing ESRI Personal geodatabases are usually encountered only in the form of Access .MDB files and only rarely using SQL Server, where SDE appears to predominate.

Manifold can store images and surfaces in SQL Server 2008 and SQL Server 2005 using Manifold's generic spatial DBMS capabilities. Manifold Enterprise Edition or above is required to connect using native connections to spatial DBMS such as Oracle Spatial, DB2 with IBM Spatial Extender, SQL Server 2008 spatial and PostgreSQL / PostGIS.

**Very Important:** Due to changes in SQL Server 2008 between pre-releases and final production form, it is very important that all work with SQL Server 2008 be done using a production version of SQL Server 2008 and the current version of Manifold. Do not use pre-release versions of SQL Server 2008 (so-called "CTP" versions) and do not use old versions of Manifold.

**SQL Server 2008 Geometry Types**

SQL Server 2008 spatial capabilities include support for two native geometry types: a planar GEOMETRY type similar to the geometry types used by other spatial DBMS products and a geodetic GEOGRAPHY type. The internal workings of either type have not been documented by Microsoft as of this writing, so Manifold uses these types the same way as it does SDO_GEOMETRY from Oracle Spatial.

That is, when data is brought in from SQL Server native GEOMETRY or GEOGRAPHY storage by importing or linking a drawing it is dynamically cast into Manifold Geometry type. When a Manifold drawing is exported into SQL Server, Manifold Geometry is automatically cast into SQL Server native GEOMETRY.

Any edits or other operations performed on SQL Server native GEOMETRY or GEOGRAPHY drawings that are linked into Manifold are dynamically manipulated so that whatever is done in the Manifold session is correctly updated within SQL Server native storage. Any re-projection required to show a linked SQL Server native GEOMETRY or GEOGRAPHY drawing within, say, a Manifold map in a different projection will automatically be done on the fly.
Note that Manifold can import or link a drawing from either SQL Server native GEOMETRY or GEOGRAPHY but that exports from Manifold will always be exported into SQL Server native GEOMETRY. Exports are not available into SQL Server GEOGRAPHY.

If you have a drawing that uses SQL Server GEOGRAPHY Manifold will be happy to use it. You can import it or link it into Manifold. If you link it into your Manifold project, Manifold will happily allow you to edit objects in it, to delete them, to create new objects and so forth. But Manifold will not create a new drawing for you within SQL Server that uses GEOGRAPHY. This limitation arises from the unique nature of the GEOGRAPHY type and may be relaxed in future versions. For the time being, it is not in any way a limitation for GIS operations since GIS data at the present time is universally planar in form, such as the SQL Server GEOMETRY type.

It is also possible that should Microsoft document either the GEOMETRY or GEOGRAPHY type Manifold will add these as intrinsic types that can be held in storage outside of SQL Server, much as can OGC WKB be used in the form of the Manifold Geometry(WKB) type.

**SQL Server 2008 Projections**

The GEOGRAPHY type always uses Latitude / Longitude. Importing or linking a drawing from SQL Server 2008 that uses GEOGRAPHY type will import or link the drawing into Manifold using Latitude / Longitude projection. Manifold recommends that all storage into GEOGRAPHY type adopt the convention of utilizing coordinates developed using the WGS84 datum.

When reading and writing spatial data stored on SQL Server 2008 spatial using Microsoft's GEOMETRY type, Manifold represents coordinate systems using EPSG codes. EPSG, the former European Petroleum Survey Group organization, has been absorbed into the International Association of Oil and Gas Producers (OGP), but the database of coordinate systems is still widely known as EPSG. EPSG comprises one of the largest vendor-neutral, unambiguous and expertly-developed sets of coordinate systems known. Using EPSG with SQL Server 2008 spatial GEOMETRY is a great way of eliminating the confusion one might encounter with proprietary methods of specifying coordinate systems.

Although EPSG codes cover very many coordinate systems, because Manifold can handle a much larger set of projections than those covered by EPSG, it is possible that a drawing in Manifold uses a coordinate system (projection) not available in EPSG. For example, EPSG does not have a code for Orthographic projection.

If a Manifold drawing uses a coordinate system not supported by EPSG, when the drawing is exported into SQL Server 2008 GEOMETRY, Manifold will re-project it on the fly into Latitude / Longitude form (using EPSG code 4326). Built-in Manifold coordinate systems include all EPSG coordinate systems so that drawings imported or linked from SQL Server 2008 GEOGRAPHY need never be re-projected for use in Manifold.

SQL Server 2008 spatial allows third parties to choose some other coding method to represent coordinate systems instead of EPSG. If some third party application has used some other coding scheme then Manifold will not be able to automatically detect the correct coordinate system in use, and the correct coordinate system will have to be specified using the Edit - Assign Projection dialog after the drawing is imported or linked into Manifold.

**Re-projection in SQL Server 2008**

Note that although Manifold will happily re-project data as necessary and can even re-project on the fly so that drawings stored within SQL Server 2008 using different projections can be used together for spatial analytics such as topology overlays, as of the present writing SQL Server 2008 does not have server-side re-projection capability. This means that the usual Manifold facilities for changing projections of drawings stored in spatial DBMS won't work if native SQL Server 2008 spatial enhancements are used. We can, however, still use Administrator Console (available in Database Administrator edition or Ultimate edition) to examine the projections in use.
Launching the administrator console we can connect to SQL Server 2008 using the data source that is configured in the example topics below.

We press the Columns button in the console’s toolbar to turn on the Projection column if it is not already visible. This reports the projection in use by drawings. We choose a drawing, a version of the standard Mexico example, that has been uploaded to SQL Server 2008 spatial storage using an unusual projection, MTM Zone 3. We double-click into the projection cell for that drawing.

The Projection dialog used with EPSG codes reports the details of the projection in use. We can scroll down to EPSG 4326 and click on it to highlight it. We can try to re-project by pressing the OK button.
However, we cannot modify the projection (coordinate system) in place.

If we want to re-project the drawing this is easy to do in Manifold: import the drawing into Manifold, re-project it within Manifold, and then export it back into SQL Server 2008.

**The Manifold Spatial Extender**

The Manifold Spatial Extender for SQL Server is a free extension module for SQL Server which provides spatial index functionality for Manifold System. The Manifold spatial extender works with all versions of SQL Server starting with SQL Server 2005. Always make a point of installing the Manifold spatial extender on the server machine whenever using SQL Server 2005 as a spatial DBMS.

Although the Manifold spatial extender works fine with SQL Server 2008 as well, normally we would use native SQL Server 2008 spatial capabilities instead of installing the Manifold spatial extender and using generic Manifold spatial capabilities.

See the Manifold Spatial Extender for SQL Server topic for details.

**Technical Note on Pre-Release GEOGRAPHY Type Coordinate Ordering**

This note applies only to users who experimented with pre-release versions of SQL Server 2008 or who are still using an older Manifold release with SQL Server 2008. If you are using a production version of SQL Server 2008 with a current Manifold release this note does not apply to you.

Experimenting with pre-release software always involves the risk that something important will change in the production version. Users who experimented with pre-release versions of SQL Server 2008 should take note of a very important change made by Microsoft in the final production version of SQL Server 2008.

During Microsoft's development of SQL Server 2008 several pre-releases, called CTP (Community Technology Preview) releases were issued during 2007 and 2008. Up until the final CTP and final release to manufacturing in 2008, SQL Server 2008 GEOGRAPHY type was unique in that it used (latitude, longitude) coordinate ordering, that is (Y, X), ordering instead of the (X, Y), (longitude, latitude) ordering that is universally used by all other spatial DBMS and GIS products and essentially all other vector and raster GIS formats, including SQL Server's own GEOMETRY type when storing Latitude / Longitude data.

Based upon feedback from beta testers and the development community, the final CTP and final production release of SQL Server 2008 changed the ordering of coordinates in GEOGRAPHY type to the industry standard, that is (X, Y), (longitude, latitude) ordering. Every application designed around pre-release versions of SQL Server 2008, including Manifold, therefore had to change accordingly.

Because Manifold supported GEOGRAPHY type since the very first spatially-capable CTP in 2007, all Manifold Release 8 builds published before the final release of SQL Server 2008 took into account the backwards ordering of coordinates in SQL Server 2008 pre-releases. When importing or linking GEOGRAPHY, all Manifold builds published up through and including July, 2008, would automatically swap coordinate ordering into the (X, Y) order expected by all other systems, including Manifold. The last Manifold build that automatically swapped coordinate ordering was 8.0.8.

All Manifold releases from August, 2008 onward were changed to follow SQL Server's adoption of industry standard ordering so that they do not swap coordinate order for GEOGRAPHY type. If you are reading this Help topic from within your Manifold installation you are operating a newer Manifold release which does not swap coordinate order in SQL Server 2008 GEOMETRY type.
Developers who were working with pre-release SQL Server 2008 versions should be aware that SQL Server 2008 GEOGRAPHY coordinate order has changed from backwards ordering in the earlier CTPs to industry standard ordering in production SQL Server 2008 and that Manifold has also changed accordingly. Developers should be aware that using a newer Manifold release with older data stored in a pre-release SQL Server 2008 database will result in backwards coordinate ordering, since the old SQL Server system was backwards but current Manifold releases no longer automatically flip the backwards ordering. Likewise, using a Manifold release issued on or before July, 2008, with the production version of SQL Server 2008 will also result in backwards coordinate interpretation, because the older Manifold release will think that the SQL Server coordinates are backwards and will swap them automatically.

The solution to avoiding all problems is very simple: use the latest Manifold release with production SQL Server 2008. Do not use pre-release SQL Server 2008 versions and do not use older Manifold releases with production SQL Server 2008.

**SQL Server 2008 Examples**

SQL Server 2008 has been used in a series of example topics illustrating work with this fine new Microsoft spatial DBMS product.

Example: Configuring SQL Server 2008
Example: Linking a Drawing from SQL Server 2008
Example: Storing a Drawing in Manifold Spatial DBMS
Example: Storing a Drawing in SQL Server 2008
Example: Tracing Virtual Earth into SQL Server 2008
Example: Storing an Image in SQL Server 2008

**SQL Server 2005 Examples**

SQL Server 2005 Express has been used in a series of example topics illustrating use of generic Manifold spatial DBMS capabilities. Many of the techniques illustrated (such as use of the Administrator Console) also apply to SQL Server 2008.

Example: Storing a Drawing in Manifold Spatial DBMS
Example: Storing an Image in Manifold Spatial DBMS
Example: Storing a Surface in Manifold Spatial DBMS

**See Also**

Spatial DBMS
Spatial DBMS Facilities

Data Storage Strategies
Database Installations
Database Administrator Edition
ESRI Geodatabases
Geometry in Tables
Linked Drawings
Manifold Spatial DBMS Facilities
Manifold Spatial Extender for SQL Server
Oracle Spatial Facilities
Project Pane - Open Data Source
Queries and Geoms
Spatial Extensions
SQL Server Spatial DBMS Facilities
Tools - Administrator Console
Tools - Database Console
Example: Configuring SQL Server 2008

The example assumes SQL Server 2008 has been installed on a machine called GL and we are working on that same machine. The default installation from SERVERS/SPLASH.HTA was used. (Screenshots were made using pre-production SQL Server 2008 software. They may be slightly different from production versions of SQL Server.)

This example shows how to configure Manifold data sources for a SQL Server 2008 installation using the Data Source dialog. It also shows how to use the Database Console to configure SQL Server 2008 for use by an explicit user and password instead of Windows integrated security.

Sometimes when working with SQL Server it is easiest to use explicit user / password pairs instead of integrated security. The classic example is using SQL Server 2008 spatial data within a Manifold Internet Map Server project, where it might be a hassle to use Windows integrated security to assure access for the IUSR_xxx account. This example shows how to set up SQL Server 2008 while providing, in passing, a detailed, step-by-step example of how various Manifold dialogs, like the Data Source dialog and the Database Console dialog, are used.

We begin by launching SQL Server Management Studio from the Windows taskbar Start button.

Using (local) for the Server name means we are working on the same machine. Press Connect.

1. Launch SQL Server Management Studio.
2. Connect to SQL Server using (local) as the Server name.
3. Press Connect.
We right-click on the server node in the Object Explorer window and choose **Properties**.

![Server Properties - GL](image)

We click on the **Security** page (in the left hand pane) and choose **SQL Server and Windows Authentication** and then click **OK**.

![Microsoft SQL Server Management Studio](image)

Back in the Object Explorer pane we again right-click the server node and choose **Restart**. We wait until the system starts and restarts the service and then we can close the SQL Server Management Studio application.

We now launch Manifold.

![Project1 - Manifold System](image)
From the main Manifold menu we choose Tools - Database Console to launch the Database Console. The Database Console allows us to browse databases and to import and link data from those databases. The lower pane of the Database Console allows us to execute queries within the database server. Now would be a good time to review the Tools - Database Console topic.

This is the first time we have used this Manifold installation, so no data sources have yet been configured. We click the [...] browse button to open the Data Sources dialog.

Now would be a good time to read the Data Source Dialog topic, as this dialog is the friend of every DBMS user. It remembers previously used data sources that we have configured, so we can easily connect to data sources with a simple point and click. It takes a moment or two to add a new data source to the dialog, but once the data source is in there we save a lot of time afterwards.
The dialog opens up with no data sources configured, since this is the first time we have used this Manifold installation and no data sources have yet been configured. We press the Add Data Source button.

That opens a typical Windows file open dialog. We choose SQL Server Data Sources as the type. Using this type means to use the native connection to SQL Server 2008 spatial DBMS.

We first will configure a data source using Windows integrated security, since we will use our status as Administrator (yes, that's how we logged in...) to configure a user, password and database within SQL Server 2008. We can then later configure a data source to use that user / password pair. Press OK.

Back in the Data Source dialog we have a new data source added to the list. We will rename it to something that later will remind us what this source is all about. We will use SQL Server - Windows Integrated Security. Press OK.
That takes us back to the Database Console, which shows the currently empty status of the server. We will now execute a series of statements in the lower query pane to configure the DBMS.

Enter the statement

```sql
create database TESTDB
```

in the lower pane and press the Run button. This executes the statement. Follow that by executing each of the following statements, pressing the Run button after each in turn:
Enter the statement

create login TEST with password='TESTPWD'

in the lower pane and press the Run button. This executes the statement.

Enter the statement

create user TEST for login TEST
use TESTDB
create user TEST for login TEST

in the lower pane and press the Run button. This executes the statement.

Enter the statement

use TESTDB
grant control to TEST

in the lower pane and press the Run button. This executes the statement.

We could, of course, if desired substitute our preferred user name, password and database name for the uppercase words in the above statements. The statements configure our SQL Server 2008 server to allow use of a user and password pair. The above statements have been broken into simpler chunks to illustration what is being done and to help avoid typos. An experienced user would execute them all together entered into the lower pane.
After executing the last statement above the Database Console will remember it in case we want to use it again or edit it. If that is not desired it is a good idea to delete the statement text so that future Database Console sessions won’t have it appear in the lower pane.

We can now configure a new data source in Manifold that connects to the server as user TEST with password TESTPWD and using TESTDB as a database. To configure and use a new data source we again press the browse button.

Once more we press the Add Data Source button.
We now choose **Use specific user name and password** and provide the name, password and database name to use. Press **OK**.

A new data source appears in the list. We rename it **SQL Server - TEST user / password** to remind us what this is about. We double-click the data source to connect using it.
Database Console is now connected using the new data source. (We've deleted the last statement issued from the query pane at the bottom.)

In subsequent examples, we will use the new data source to export data into SQL Server 2008, beginning with the Example: Storing a Drawing in SQL Server 2008 topic.

See Also

Data Source Dialog
Example: Storing a Drawing in SQL Server 2008
Spatial DBMS Facilities
SQL Server Spatial DBMS Facilities
Tools - Database Console
Example: Storing a Drawing in SQL Server 2008

This topic shows a simple example of exporting a drawing into SQL Server 2008. It uses the data source created in the Example: Configuring SQL Server 2008 topic.

Launch Manifold and open the Mexico drawing used in numerous examples in the Examples chapter.

To export the drawing we choose File - Export - Drawing and in the type box choose Data Sources ().
In the resulting Data Source dialog we click the user / password pair data source set up in the prior example. This connects to the server using a specified user name and password and desired database. Press OK (or simply double-click the data source).
In the resulting Export Drawing dialog we use defaults, checking to make sure the Type is SQL Server so that we use native SQL Server 2008 spatial storage. By default, all fields (columns) will be exported except intrinsic fields. The drawing will be uploaded into the DBMS using standard Manifold geometry types and a spatial index will be created. Press OK. Done!

Important: When exporting a drawing to a spatial DBMS it is best if the drawing does not contain a column called OID. This name is used by Manifold as the default name for the object identity field. If we like, we can still export a drawing that contains such a column but then we will have to remember in the export dialog to change the name of the identity field to something else.

While that is certainly easy enough to do, it is also easy to forget to do in which case the export will fail and we will wonder why. Also, it is useful to always assume that an OID field is the identity field. Therefore, it is strongly recommended to treat this field name as a reserved name.

We've now saved the drawing into SQL Server 2008 spatial storage. If we want to use the drawing, we can import it from the server or link to it using the Tools - Database Console dialog as shown in topics such as the following Example: Linking a Drawing from SQL Server 2008 topic.

See Also

Data Source Dialog
Example: Configuring SQL Server 2008
Example: Linking a Drawing from SQL Server 2008
Example: Linking a Drawing from SQL Server 2008
Example: Storing a Drawing in Manifold Spatial DBMS
Example: Tracing Virtual Earth into SQL Server 2008
Multi-User Editing of Linked Drawings
Spatial DBMS Facilities
SQL Server Spatial DBMS Facilities
Tools - Database Console
Example: Linking a Drawing from SQL Server 2008

This example shows how to link a drawing stored in SQL Server 2008, using the example drawing stored in the Example: Storing a Drawing in SQL Server 2008 topic. In addition, it shows how the drawing can be used with transform operators, dynamically working with “live” data that is stored within SQL Server 2008 geometry. This example assumes data sources have already been configured as set forth in the Example: Configuring SQL Server 2008 topic.

Recall that we started with the example Mexico drawing seen above, and exported it into SQL Server 2008 spatial storage.

Launch a new Manifold session and launch the Database Console. The Database Console remembers the last-used data source for our convenience, so all we need do is press the Refresh button to refresh the connection to the DBMS server.
The Database Console shows the uploaded drawing with a drawing icon. It also shows the database table that contains the drawing as a table, in case we just want to import or link a table. If we want a more elegant presentation for novice users we can use Administrator Console to assign friendly names.

To link the drawing we click on it to highlight it and then press the Link button.

The Import / Link Options dialog opens to give us the option of linking the entire drawing or just a portion of it. The ability to link only those objects within a given area of interest allows us to use the power of SQL Server to store virtually limitless drawings while only working with a manageable portion of them within a given area.

The Version choice specifies what field to use within the drawing to reconcile multi-user edits. Since more than one Manifold user on more than one different machine could simultaneously link and edit this drawing, Manifold provides a way to resolve editing conflicts using a Version column. See the Multi-User Editing of Linked Drawings topic for details and an example.

We choose the Version column and press the OK button.
A new drawing appears in the project pane together with the drawing’s table. A database cylinder appears in the drawing and the table icons to indicate this component is linked from an external source.

If we open the drawing we see that it is, indeed, the example Mexico drawing exported into SQL Server 2008 spatial storage.
If we open the drawing’s table we see that all the attribute data for each drawing object (the various provinces of Mexico) have been stored with the drawing.

Like any linked table the data is immediately available for use within Manifold, just as if it was a local component stored within the project. For example, we can work with the data in the drawing’s table.

We can click on a column head to sort the column by ascending order, right-click on the column head for a variety of choices or simply use a shift-click on a column head to sort by descending order. In the illustration above we have shift-clicked on the column head giving population for each province to sort the table from the most populous province to the least populous.

We can make selections in the table as well. Let’s first change the selection mode to Select Add. See the Selection topic for an introduction to selection modes.
We can now click on the row handle to the left of each record to select that record. In the illustration above we have selected a few records in the drawing's table based on population values that have caught our eye.

As we select records in the table the corresponding objects will also light up in red selection color in the drawing window. Manifold provides select-back, whereby a selection made in any open window will be seen in corresponding objects in any other open window. We can select an object in the drawing and see its record selected in the table as well or vice versa.

With the focus on the drawing window, let's do something dramatic and press the Delete key to delete the selected objects.
Manifold helpfully asks us to confirm deletion. We press Yes.

Adios, provinces! Those objects have now been deleted from the drawing. Note that since this is a linked drawing that shows live, dynamic data directly from SQL Server 2008 spatial storage, we have truly deleted the objects from the SQL Server storage. If anyone else was working with the same drawing in a different connection to the same database, they would also see those objects disappear.

Since we would like to keep this particular example drawing whole for the rest of this example, we will use everyone’s favorite Manifold feature: undo. We click on Edit - Undo to undo the last action. Manifold provides single-step undo, an extremely useful feature at times.
Like magic, the deleted objects are restored. Note that the objects are shown in selection color. If we still have the drawing's table open in a different window we would see the associated records re-appear.

Let's shift gears a bit and do something with the linked drawing that shows how geometry from SQL Server 2008 can be used to do analytics.

With the focus on the drawing window we choose Edit - Select None to clear the selection.

In the Transform Toolbar we choose [All Objects] and the Centroids (Weight) transform and press Apply.
This transform operator creates a point object at the centroid by weight, that is, at what would be the balance point if the area were cut out of stiff paper, and selects the centroid points.

Note that since this is a linked drawing that is working with a live, dynamic connection to the drawing stored within SQL Server 2008, we have actually created new point objects within the SQL Server 2008 spatial storage.

[Although we don't use this characteristic, when Manifold creates a centroid for each area object, those centroid points each inherit the data attributes of the originating area. We can control how that inheritance occurs by using Transfer Rules.]

We don't want to clutter our drawing with point centroids, so we will choose Edit - Cut to cut them out of the drawing (we could also use the standard Windows CTRL-X shortcut).

We then right-click into the project pane and choose Paste.
Manifold® System Release 8.00 User Manual

Manifold cuts the centroid point objects out of the linked drawing, removing them from SQL Server 2008 spatial storage, and places them in a new drawing created in the project pane. This new drawing is a local drawing, stored within the Manifold project file. As such it is shown using a drawing icon and a table icon that do not have a database cylinder in the icons.

We right click on the new drawing and choose **Rename** and change the name to **Centroids** to make it easy to recognize what is in that drawing. Manifold by default will helpfully rename the drawing’s table for us as well.
If we open the new Centroids drawing we can see that it contains the centroid points that were cut out of the Mexico drawing. The Centroids drawing uses the same projection as the Mexico drawing, basing that upon the projection used by the objects that were cut and pasted to create the new drawing.

Let's change the projection of the Centroids drawing. This is not an example about projections, so we will go quickly.

We first use the Edit - Assign Projection dialog to verify that the Centroids drawing is indeed intended to be in Latitude / Longitude. We next launch the Edit - Change Projection dialog to change the projection.
In the Edit - Change Projection dialog we drill down and choose Orthographic as our desired projection. This is the classic "view from space" projection and has the advantage of using meters as a unit of measure, so it is very convenient for measurement calculations.

We press the Suggest button to get Manifold to suggest some reasonable values for the central latitude and longitude and then we press OK.
There is not much visible difference between the original projection in Latitude / Longitude projection and the new display using Orthographic.

Let's now use the transform toolbar again to create buffer zones.

With the focus on the Centroids drawing we use the Buffers transform to create buffers about each object that use a radius of 200,000 meters (200 kilometers). Press Apply.
Manifold creates circular buffer areas around each point, with each buffer area inheriting the data attributes from the centroid point from which it was created. Each newly-created buffer area is also selected, making it convenient to further use the areas in subsequent operations.

We will **CTRL-X** to cut the buffer areas and then **Paste** them into the project pane as a new drawing, renaming the new drawing to **Buffers** to remind ourselves what that drawing contains.

Opening the new **Buffers** drawing we see that it does indeed contain the circular buffer areas we created from the centroids.

We now choose **Create - Map** from the project pane toolbar to create a map using the **Buffers** drawing as the first layer. We first create the map using the **Buffers** drawing because we want the new map to use the same
projection. Orthographic as the Buffers drawing (which is in Orthographic because the buffer areas that were cut from the Centroids drawing and pasted to create the Buffers drawing were in Orthographic). Press OK.

We open the new map and we drag and drop the Mexico drawing from the project pane into the map. This adds the Mexico drawing to the map as a layer. We drag the Buffers layer tab to the left of the Mexico tab so that the Buffers layer appears above the Mexico layer.

Even though the Mexico drawing is in Latitude / Longitude projection it is re-projected on the fly to use the Orthographic projection used by the map. Re-projection on the fly can take some computational time in the case of large, complex drawings, so it is wise when showing layers in a map to use the projection of the largest, most complex layer for the map’s projection as well. These are such small components that rendering will be fast in all cases.
Too much opacity in the **Buffers** layer does not an elegant presentation make, so we will right click on the **Buffers** layer tab, choose **Opacity** and set the opacity to about 50 percent. That allows more of the **Mexico** layer underneath to show through to provide better visual context. Manifold has endless such visual capabilities. Although this is mainly a database storage example, it is still useful to employ such capabilities for a better visual presentation.

We next will do a slightly more sophisticated operation that is not shown in step-by-step detail: we will dissolve all of the different buffer areas into a single area object.

We do this by opening the **Buffers** drawing's table, selecting all records, changing the value in one of the fields (such as the **Auto** field) to 1, which is a shortcut way of changing that value for that column for all of the records.
We then click on the map to move the focus back to the drawing layer and choose **Drawing - Dissolve**, dissolving on the **Auto** field. Manifold will dissolve all of the records with the same value in that field into the same object, in this case making one object out of all the buffer areas.

Note that the resulting object, as seen above is a single, branched object. That is, even though it appears to be at least two separate areas both of those apparently separate areas are part of the same complex, branched single area object. Manifold can automatically handle branched objects such as areas that contain holes or are single objects with multiple islands.

The above is a lot of preparation just to get a layer that we will now use to "cookie-cut" our **Mexico** drawing.

In the transform toolbar, we use the Clip with (Intersect) transform to clip all objects in the **Mexico** drawing with all objects in the **Buffers** drawing. Press **Apply**.

Manifold uses the buffer area to cut all of the area objects in the **Mexico** drawing, deleting any portions of area objects in the **Mexico** drawing that do not fall within a buffer zone. This is a very common operation in GIS.

For example, suppose the buffer area represents the coverage radius from radio antennas, as in a cell phone application. If we had an antenna at the centroid of each province in Mexico and the range was 200 kilometers from each antenna, the buffer zone shows us the coverage attainable. The clip transform shows the land area that is reachable from all antennas.

Note that even though the objects in the **Mexico** drawing are in a different projection from the **Buffers** drawing the operation still works. Manifold will automatically re-project on the fly as necessary when doing spatial analytics.

Note also that even though the objects in the **Mexico** drawing are dynamically linked into Manifold from SQL Server 2008 spatial storage the transform still works. The objects are automatically edited in place within SQL Server by the transform.
We can open the **Mexico** drawing in its own window to see the drawing on its own, seen in the **Latitude / Longitude** projection used by this drawing.

We choose **Edit - Select None** to deselect all objects to get a clear view of the objects as edited.
We can select just one of the objects to see the intricate shape automatically created when the complex buffer zone object was used to slice through the also complex shapes of Mexican provinces.

Note from this view that what appears to be a circle when viewed from space or when computed on the basis of radius in a projection that preserves linear measurements, like Orthographic, does not appear as a circle given the distortions of a cylindrical presentation like Latitude / Longitude. The cuts in the Northwest part of Mexico appear slightly ellipsoidal when viewed in Latitude / Longitude: attempting to "draw a circle" in a Latitude / Longitude view will not generally result in a true circle upon the Earth's surface.

**See Also**

Data Source Dialog
Example: Configuring SQL Server 2008
**Example: Storing a Drawing in Manifold Spatial DBMS**
**Example: Storing a Drawing in SQL Server 2008**
Example: Tracing Virtual Earth into SQL Server 2008
Multi-User Editing of Linked Drawings
**Spatial DBMS Facilities**
SQL Server Spatial DBMS Facilities
Tools - Database Console
Example: Tracing Virtual Earth into SQL Server 2008

This topic continues the series of SQL Server 2008 examples set forth in the SQL Server Spatial DBMS Facilities topic. We use data from Microsoft Virtual Earth to create a drawing within SQL Server 2008 showing buildings in the Microsoft Redmond campus and then we edit that drawing to show how drawings linked from SQL Server 2008 into Manifold maintain their dynamic links back to SQL Server spatial storage. This example assumes data sources have already been configured as set forth in the Example: Configuring SQL Server 2008 topic.

We begin this example by opening the US Base Map.map project from the Exporting KML to Google Earth tutorial topic. That project has two handy linked images, one from Microsoft Virtual Earth satellite imagery and the other from Microsoft Virtual Earth road map imagery, and also has a useful map of the US as boundaries. We are mainly interested in the Virtual Earth layers. This example assumes, of course, that we are connected to the web so that Manifold facilities such as image server module interfaces, will operate correctly.

We delete all the components from that project we don't need and we save it under a different name. This will be our new project for creating a drawing that shows buildings in the Microsoft Redmond campus.

We begin with a project that has two linked images and a drawing. We won't use the USA drawing but it is there in the project should we need it. We create a map from the Virtual Earth Satellite Image and we open the map.

Wow! As always, the sheer breadth and quality of Virtual Earth imagery never fails to impress. This particular image was linked using the extents of the entire US plus parts of Mexico and Canada. The amazing thing about the image before us is that we can zoom into any part of it and Manifold will arrange to stream in more detailed image tiles to the limit of resolution supported by the host image server. Considering that the host image server is Microsoft Virtual Earth that ultimate detail is impressive indeed. We can zoom into this image to see virtually any part of the US in astonishing detail.
Our objective is to zoom into that part of Redmond hosting the main Microsoft campus, but it is not so easy to see where Redmond is from a satellite view.

To provide better orientation, we will drag the Microsoft Virtual Earth street map image into the map and then use the Zoom Box tool to zoom into the Seattle area.
Note that the magic of Manifold image server technology connected to Virtual Earth automatically fetches more detailed views. Knowing that Redmond is somewhere near Bellevue we will zoom again into the Seattle area near Bellevue.
There's Redmond... let's zoom closer.
At this point we know we are looking at Redmond, but we are darned if we know where exactly Microsoft is in this display. To get a fix on the Microsoft campus, we will use Manifold's built-in geocoding capability (available if we have the Geocoding Tools extension installed or any edition from Universal edition on up).

We choose **Edit - Go To**, choose **Address** in the **Go To** box and give the published address of Microsoft Corporation. Press **OK**.

That's better. Let's zoom further into the Microsoft campus.
At this point we have zoomed far enough into the Virtual Earth street image that we can see individual Microsoft buildings. Time to switch to the satellite image.
We do this by double-clicking on the Virtual Earth Street Map Image layer to turn it off, and clicking on the Virtual Earth Satellite Image layer to make it the active layer. We zoom further into the image.
That's a good view of much of the main Microsoft campus in Redmond. Let's zoom even further into the image, to the point that an individual Microsoft building fills the screen.
Very good. Now we will switch gears a bit and create a drawing into which we can draw areas that trace over the shapes of individual buildings.

To do so, we right click onto the Virtual Earth Satellite Image tab and choose Add - New Drawing from the context menu.

The Create Drawing dialog allows us to name the new drawing and to provide any descriptive notes we would like. We enter the name Buildings and press OK.
A new tab appears in the map for the new drawing.

In the project pane, we see that a new drawing, called **Buildings**, has been created, together with the drawing’s table.
If we open the drawing's table we see that it is empty and consists of just one column, the built-in ID column that stores the object ID for any objects added to the drawing. We will add another column. To do so, we right-click on the column head.

From the context menu we choose Add - Column.

In the Add Column dialog we provide a name for the new column (we will call it Buildings) and for the type of the column we choose Text (ANSI, variable-length) from the seemingly endless list of different database types offered by Manifold in the Type box pull-down menu. Press OK.

Very good. We now have a new column in the drawing's table. So far there is nothing in the table because we have not added any objects to the drawing.

One more thing: because this is an example and we want to show off a bit, we will change the new drawing's projection. When a new drawing is created in a map it inherits the projection of the map, in this case the same Mercator projection used by the Virtual Earth layers from which the map was created.
Changing the projection of the Buildings drawing is easy. First, we right click on the Buildings drawing in the project pane and choose Assign Projection to verify that the starting projection is what we intend it to be. This is a "belt and suspenders" safety check by Manifold applied to all components the first time they are used in a projections context.

Next, we right click on the Buildings drawing in the project pane and choose Change Projection to get the dialog shown above. In that dialog we choose Latitude / Longitude projection and press OK. That changes the projection of the drawing to Latitude / Longitude projection.

When the projection of a layer in a map is changed the map will re-sync itself as it re-projects the component on the fly to display it in the projection used by the map. We click the Back button to get back to the same zoom and pan used earlier.

There is one more housekeeping thing we will do before launching into a wonderful and productive editing session. We will turn on Instant Data mode.
To do this, we choose **Edit - Instant Data**. This sets a highly useful editing mode so that whenever we draw a new object into a drawing Manifold will automatically pop open an editing dialog to enable us to add field values for that object. This is one of those deals that sounds like a totally nerd-alert, dry feature but in actual practice is one of those things you cannot live without once you taste it.

With the focus on the **Buildings** drawing layer in our map, we click the Insert Area command to use that tool.

To use this tool we click on the outlines of the area desired. In this case we click along the edges of the building over which we want to create an area. This is a good example of tracing, whereby we create new map objects by tracing in a drawing layer above an image layer.

When we are finished drawing the area we **right click** to indicate we are done.
A new area appears in the drawing and because we have Instant Data mode on a dialog pops open to enable us to enter attribute data for that object. We enter 21 into the Buildings field because we know this is building 21 in the Microsoft campus. Press OK.
A new area object is created in the drawing.

If we open the drawing's table we see that a new record has been created for that drawing object, with the value of **21** for the **Buildings** field.

Let's pan over to a nearby building and trace over it to create another area object.
When we are done the Instant Data mode automatically pops open the dialog with the cursor pre-positioned in the last field edited with the last-used data entered, ready for us to modify it or enter new data.

Instant Data makes it extremely efficient to enter large numbers of new objects together with data attribute information for each new object. It is the first choice for data entry when recording assets or other repetitive information. For example, a crew can use it together with Manifold's GPS Console capability to record the location of thousands of assets, such as fire hydrants or electrical transformers or even 911 addressing in a day.

We enter **35** for this building and press **OK**.
In the drawing’s table we see that we now have two records for the two objects in the drawing.

Zooming out a bit in the map we see there are now two area objects drawn into the Buildings drawing.

Let’s switch gears and export the Buildings drawing into SQL Server 2008 spatial storage. We can’t export a drawing into SQL Server unless it contains some objects. We have created two objects in this drawing so we can now export it into SQL Server. After export, we will link the drawing back into our Manifold project and continue editing it “in place” within SQL Server.

To export the drawing, we right click onto the Buildings drawing in the project pane and choose Export.
In the resulting dialog we choose **Data Sources** () as the type to which this drawing will be exported.

That launches Manifold’s Data Source dialog, which helpfully remembers data sources that have been previously configured. We have two data sources, configured in the Example: Configuring SQL Server 2008 topic. We choose the data source using an explicit user login and password since that connects to the database used in previous examples.
In the resulting **Export Drawing** dialog we verify that the **Type** is **SQL Server** and otherwise accept defaults. Press **OK**.

The drawing is now exported into SQL Server 2008 spatial storage. For the rest of this example we will use the drawing resident within SQL Server.

To do that, we will first right click on the **Buildings** drawing in the project pane and delete it from our project. This also deletes it as a layer from the map. Next, we will link the drawing back in from the database. We will do so by launching Tools - Database Console.
The database console helpfully remembers the last-used data source. We press the Refresh button to refresh the connection.

After refresh, the database console lights up with a listing of all the components in this database. As we can see, there are plenty. For now, we have not used Administrator Console to assign friendly names so they are displayed in their native SQL Server object names. Manifold will helpfully put a drawing icon next to those database objects that it knows contain drawing geometry.

We click on the database object that is obviously our Buildings drawing uploaded in the prior step and we press the Link button to link it into our project.
The link options dialog allows us to choose to use all objects or just some area of interest subset. We use all objects since there is hardly anything to this drawing. If we wanted to grab just a manageable subset of some terabyte-sized drawing stored in SQL Server we would no doubt use some area of interest subset.

Manifold creates a version column by default to support multiuser editing of linked drawings. We make it a habit of using that column just in case someone else connected to the DBMS also wants to simultaneously edit the same drawing. Press OK.

A new drawing appears in the project pane. The drawing’s icon incorporates a database cylinder graphic to indicate this is a drawing linked into the project from an external data source.

If we open the drawing we see that it contains two area objects. They look a bit strange because the drawing uses Latitude / Longitude as its native projection, and that projection shows considerable distortion in higher latitudes like those where Redmond is located.
If we open the drawing's table we see that it does indeed contain all the data attributes uploaded into SQL Server.

However, our task is not to look at the drawing on its own but to work with it in a layer above the Virtual Earth imagery we were using earlier. We therefore drag and drop the linked drawing from the project pane into the map. It appears as a layer, re-projected on the fly into the Mercator projection used by Virtual Earth and the map. Note that the objects overlay perfectly, as we would expect them to do.
The gray default formatting of new objects is getting boring, so we will format the area background color for the Buildings layer to be bright yellow.
The interesting thing about the Buildings drawing layer is that it is a "real time" connection to the drawing stored within SQL Server. If we edit the drawing, we are dynamically changing the data in SQL Server. Although the layer seems local, it is a visual means of editing the contents of our database with none of the hassles of manually specifying coordinates or command-line interaction.

For example, we can zoom into a different building and create another area traced over it.

When we finish the object, because Instant Data mode is still on a dialog pops open to enable us to enter data for this object as well. We enter building 34.
Zooming out, we see that we now have three buildings in this drawing.

If we click open the drawing’s table (also linked in automatically from SQL Server storage), we see that we have data for three objects in the table.

What is also interesting about the drawing’s table is that it too is an interface that we can use to manage data within SQL Server. For example, suppose we want to add a new column to the table?

That is easy to do by right clicking on the column head and choosing Add - Column from the context menu
In the **Add Column** dialog we provide a name for the new column (we will call it **Cafeteria**) and for the type of the column we choose **Text (Unicode, variable-length)** just to show that Manifold can handle Unicode text as well as ANSI text. Press **OK**.

<table>
<thead>
<tr>
<th>ID</th>
<th>DID</th>
<th>Buildings</th>
<th>Cafeteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>108</td>
<td>1</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td>109</td>
<td>2</td>
<td>35</td>
<td></td>
</tr>
<tr>
<td>110</td>
<td>3</td>
<td>34</td>
<td></td>
</tr>
</tbody>
</table>

A new column appears in the table. We have actually added a new column within the table structure in the SQL Server 2008 database as well. Manifold has extraordinary capabilities for managing tables within databases, better than most DBMS products do.

Let's trace one more building. Back in the map window we use the **Insert Area** tool to create an area for another building.
When we finish the new area the instant data dialog automatically pops open as before, but this time it shows the new Cafeteria field as well. The cursor is helpfully positioned within the last field used, ready to go.

We enter 36 for the building and Y for the cafeteria field, intending to indicate this building has a cafeteria. Press **OK**.
The new area appears in the drawing. It uses the formatting we have specified for that drawing.

<table>
<thead>
<tr>
<th>ID</th>
<th>DID</th>
<th>Buildings</th>
<th>Cafeteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>108</td>
<td>1</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td>109</td>
<td>2</td>
<td>35</td>
<td></td>
</tr>
<tr>
<td>110</td>
<td>3</td>
<td>34</td>
<td></td>
</tr>
<tr>
<td>111</td>
<td>4</td>
<td>36</td>
<td>Y</td>
</tr>
</tbody>
</table>

If we open the drawing's table we see that a record for the new object, complete with attributes as we've entered them, has appeared.
Zooming out, we can see we now have four new area objects in this drawing. The last two have been added using the linked drawing from SQL Server. The edits we’ve made in this interactive Manifold session have applied in real time to the data set stored in SQL Server.
All Manifold capabilities (thousands of them) apply to the use of such linked drawings from SQL Server 2008. For example, in the illustration above we've created a labels layer that prints the Microsoft building number above each building.

We have also used thematic formatting to color the buildings automatically by whether they have a cafeteria or not. Green indicates a cafeteria. Thematic formatting and labels took, literally, less than a minute to do.

Many, many other Manifold facilities are open to us: we can create a web site in seconds that provides dynamic access to the SQL Server data, or we could publish the contents as a PDF. Although this topic uses many illustrations, it can be done by a semi-experienced Manifold user much faster than the time required to read through it. An experienced person could create this project in just a few minutes.

What is remarkable is that bringing together technologies like Virtual Earth and SQL Server and Manifold together makes it extremely easy to do what used to be very difficult and error-prone.
Consider, for example, the illustration above from the Microsoft web site which shows inaccurately the position and orientation of Building 36. In just a few short minutes we have created a more accurate rendition. Although we did not take great care to draw areas accurately, Manifold has many facilities (such as scrolling in dynamically while creating a new object) that make it fast and easy to draw accurately.

See Also

SQL Server Spatial DBMS Facilities
Example: Configuring SQL Server 2008
Example: Linking a Drawing from SQL Server 2008
Example: Storing a Drawing in Manifold Spatial DBMS
Example: Storing a Drawing in SQL Server 2008
Example: Tracing Virtual Earth into SQL Server 2008
Example: Storing an Image in SQL Server 2008

Data Source Dialog
Example: Storing a Drawing in Manifold Spatial DBMS
Example: Storing a Drawing in Oracle
Manifold Spatial DBMS Facilities
Multi-User Editing of Linked Drawings
Spatial DBMS Facilities
Tools - Database Console
Example: Storing an Image in SQL Server 2008

This topic continues the series of SQL Server 2008 examples set forth in the SQL Server Spatial DBMS Facilities topic. In this example we store a large image in SQL Server 2008 using Manifold spatial facilities. We then link it into a project to show the outstanding speed and performance obtained when even large images are stored in SQL Server 2008. This example assumes data sources have already been configured as set forth in the Example: Configuring SQL Server 2008 topic.

SQL Server 2008 spatial enhancements do not include a raster image type analogous to Oracle GeoRasters as provided within Oracle’s flagship Oracle Spatial product. However, SQL Server 2008 does a masterful job of storing binary data and when that binary data is organized as geospatial image data using Manifold spatial DBMS capabilities, the result is that we can store very large images within SQL Server 2008, preserving full projection information, and utilize them as if SQL Server 2008 had its own native geospatial image storage type.

This example uses SQL Server 2008; however, because standard Manifold spatial DBMS capabilities are used it could be done using almost any DBMS, such as SQL Server 2005, ordinary Oracle distributions, PostgreSQL/PostGIS or MySQL. It does not require use of the Manifold spatial extender, as the native facilities of SQL Server are used to manage the binary data stored by Manifold.

We will begin with an approximately 600 MB image that shows the Montara Mountain region of California. In the upper right is San Francisco International Airport on San Francisco Bay. At the lower left is the fishing port at Half Moon Bay on the Pacific Ocean. The long, diagonal lake and valley running from the center top to the center right just below and to the left of the settled area near the airport is the valley of the San Andreas fault, the rupture of which in 1906 leveled San Francisco by earthquake and fire.

This is an ordinary RGBA image kept within a Manifold project. It was created by opening the sample Montara Mountain surface used in many example topics, linking in an image from a Microsoft Virtual Earth Image Server and then downloading image tiles to approximately 0.3 meter resolution and unlinking the image to create a local
image. That is an almost completely automatic process within Manifold, easily accomplished with just a few mouse clicks on menu commands. Manifold automatically maintains projections throughout that process, so the image retains the same Universal Transverse Mercator projection originally used by the Virtual Earth image server.

To export the image we choose `File - Export - Image` and in the type box choose `Data Sources ()`.

In the resulting Data Source dialog we click the user / password pair data source that has already been set up in the Example: Configuring SQL Server 2008 topic. This connects to the server using a specified user name and password and desired database. Press `OK` (or simply double-click the data source).
The only Type available is Manifold geospatial image type, since SQL Server 2008 does not have its own image type akin to Oracle GeoRaster. The Manifold spatial database image storage mechanism automatically captures all projection information, so the image will be stored in the Universal Transverse Mercator projection it uses.

The Create Pyramids option is a very important option because it automatically creates intermediate views that enable very fast display of the image whether it is zoomed in or zoomed out.

The Tile size is automatically estimated by default and should not be changed except by experts. Manifold will automatically store the image as tiles to enable faster and more flexible data retrieval and will automatically recompose the image on the fly. The use of tiles is an internal storage mechanism that is completely invisible to users.

The Compression option likewise should not be changed. Manifold will automatically use lossless compression algorithms to reduce the space required for large images within the database. The reduction in space increases performance as well, because modern processors can decompress data faster than disk drives can fetch larger data.

Press OK. Depending on the speed of our computers, a few minutes will be required to store the image into SQL Server 2008. We can now launch a new Manifold session to illustrate how to link the image from SQL Server into Manifold.

We begin by choosing Tools - Database Console to launch the Database Console, pressing the Refresh button to refresh the connection to the data source.
Click on the image and click the Link button. [Note that if we prefer more friendly names in our database view we could use the Administrator Console to assign them.]

Almost instantly, a new linked image appears in the project pane. The database cylinder in the icon indicates the image is linked from an external source.
If we open the image we can see that it is indeed the image originally exported into SQL Server 2008. We can use the *Zoom Box* tool to zoom into the Northeast corner of the image.
Zooming occurs virtually instantly. We can zoom again for an even closer view of San Francisco Airport.
The image is so detailed that if we were to zoom to maximum resolution we could easily see every vehicle and in many cases individual people throughout many square miles of the San Francisco peninsula from the airport to Half Moon Bay. Despite the large size of the image, the power and speed of SQL Server together with Manifold are so fast that we can pan and zoom to any desired view virtually instantaneously.

Notes

Manifold spatial DBMS storage of images is so fast that if we have a fast DBMS like SQL Server 2008 available for image storage there is no gain in using compressed image technology like ECW.

For that matter, the technology in this topic scales so well that if we have the database server space and performance available to handle really large data, there is no need to manually chop up very large images into image libraries or other means of assembling larger images from smaller files. The performance available by combining Manifold with a fast DBMS allows a change in strategy for handling large files as compared to earlier technology.

For example, many government agencies and jurisdictions, from small cities to states and countries, are now compiling collections of very detailed aerial imagery. Such collections of images are often available to the public, if the public is able to handle the very large numbers of very large images involved. Such collections often include hundreds of images that are each hundreds of megabytes in size.

With older technology we might have had to deal with many large images by keeping the images separate so that each could be displayed with reasonable performance. Manifold storage of images in DBMS is so fast that an entire collection of hundreds of images could be combined into a single 100 GB+ image stored within a DBMS and Manifold would be able to display it almost instantly, using the spatial structures maintained by Manifold to fetch the right tiles for the right intermediate or full-resolution data for whatever view is required.
As a practical matter, most users will still keep some reasonable upper limit to the size of any one image in order to facilitate interchange and uploading into the DBMS, but there is no longer much of an upper limit in most applications to the size of images that, once stored in the DBMS, can be displayed within Manifold.

**Historical Note**

Gaspar de Portola (1717? - 1784?)

To the West of the Northern portion of San Andreas Lake stands Sweeney Ridge, upon the 1200 foot summit of which the expedition led by Gaspar de Portola on November 4, 1769, became the first Europeans to see San Francisco Bay.

For hundreds of years explorers had sailed past the rocky, narrow, difficult-to-see entrance to the greatest bay and natural anchorage in California without realizing the bay existed. For example, in 1580 Sir Francis Drake sailed past the entrance to San Francisco Bay and even put in for repairs to his ship, the Golden Hind, on the California coast just a few miles from the bay. Portola's overland expedition had walked past his earlier discovery of Monterey to find at last the great bay by land.

An intrepid explorer and able leader, Portola led numerous expeditions throughout Alta California, playing a key role in the founding of San Diego and Monterey and serving as Governor of Las Californias.

**See Also**

*SQL Server Spatial DBMS Facilities*
- Example: Configuring SQL Server 2008
- Example: Linking a Drawing from SQL Server 2008
- Example: Storing a Drawing in Manifold Spatial DBMS
- Example: Storing a Drawing in SQL Server 2008
- Example: Tracing Virtual Earth into SQL Server 2008
- Example: Storing an Image in SQL Server 2008

*Data Source Dialog*
- Example: Storing an Image in Manifold Spatial DBMS

*Manifold Spatial DBMS Facilities*
- Spatial DBMS Facilities

*Tools - Database Console*
Manifold Spatial Extender for SQL Server

The Manifold Spatial Extender for SQL Server is a free extension module for SQL Server which provides spatial index functionality for Manifold System. Manifold Spatial Extender works with all versions of SQL Server starting with SQL Server 2005. Using Manifold Spatial Extender on SQL Server 2005 Express requires SQL Server 2005 Express SP2. The extender operates via the use of the common language runtime (CLR) integration feature of SQL Server.

SQL Server 2008 spatial users should note that the Manifold Spatial Extender is not required to utilize SQL Server 2008 spatial capabilities. The Manifold Spatial Extender is an add-in that allows Manifold-managed spatial infrastructure for SQL Server 2005 to operate with especially high performance. SQL Server 2008 spatial users should use native SQL Server 2008 spatial capability.

Download the Manifold Spatial Extender from links in the Manifold Updates page at http://www.manifold.net/updates or from other links on the manifold.net web site. The download provides the extender and a small installation utility to simplify installation of the extension and configuration of SQL Server.

The utility installs on the machine on which SQL Server runs. It works as an extension to SQL Server to provide more effective management of spatial indices using SQL Server resources. It does not provide built-in spatial operators to SQL Server or otherwise give SQL Server the ability to function as a spatial DBMS if Manifold is not used as a client.

The spatial extender automatically works together with Manifold as a client when Manifold connects to SQL Server via OLE DB. Manifold will automatically recognize the presence of the spatial extender and will use it to create and manage spatial indices when drawings are exported to SQL Server. A key benefit of the spatial extender is that it will automatically grow spatial indices as required. This makes it possible to add new objects to a drawing beyond the original extents of the drawing when it was uploaded.

Exporting a drawing using a Manifold drawing type with the geometry type set to Geometry or Geometry (WKB) to an instance of SQL Server with the Manifold spatial extender installed creates a server-side index managed by the extender. Using a server-side index provides for slightly better performance on reads, significantly better performance on writes, and makes writes more robust. The spatial extender also allows SQL Server to operate more efficiently in a spatial server / Manifold client relationship, so it lowers load on the server as well.

The Manifold spatial extender is a free utility made available to Manifold System licensees at no charge. It is nonetheless licensed software that is covered by the Manifold EULA. You may install it on as many machines as you like so long as your intent is to use it primarily with Manifold System.

SQL Server Alternatives

The purpose of the spatial extender is to give SQL Server 2005 users the fastest possible spatial indices when used with Manifold System. This will allow users committed to SQL Server to get started today with speedy spatial technology in a way that will in the future allow easy migration to SQL Server 2008 and will (hopefully) help avoid fragmentation of the pre-2008 SQL Server user base into many different, incompatible spatial extenders that collectively will be difficult to migrate to SQL Server 2008.

Note that there are two similar but significantly different Manifold approaches to providing spatial indices for SQL Server that are managed by Manifold (as opposed to using built-in Microsoft spatial facilities as are provided in SQL Server 2008 spatial). The first is using generic spatial indices as can be done with almost any DBMS. The second is installing the Manifold spatial extender in SQL Server and then using spatial indices managed by the spatial extender.

Both can even be used at the same time, although that would be a rare and unusual situation since the spatial extender swings into action automatically with Manifold once it is installed. However, if we began work with, say, SQL Server 2005 using generic spatial indices and then later on installed the spatial extender and uploaded some more drawings, in that case Manifold would use either the generic spatial index originally created or the new spatial index managed by spatial extender as the case may be for individual drawings. To avoid any guesswork as to what spatial index is being used it is strongly recommended that the spatial extender be installed on SQL Server 2005 installations and used from the beginning for all Manifold work.

As a practical matter, the spatial extender improves performance so much that it should always be used in SQL Server 2005 installations in preference to using generic spatial indices without the spatial extender. The manifold.net team knows of no disadvantage to using it. It is small, robust, very fast, perfectly integrated with Manifold and, of course, free of charge.
The spatial extender can work with Manifold drawings uploaded using either Manifold Geometry or OGC Geometry (WKB) geometry types. Because Manifold Geometry is a higher performance and more robust type than OGC Geometry (WKB), if direct interoperability with WKB applications is not a requirement then Manifold Geometry should always be used as the geometry type.

See Also

Geometry in Tables
Spatial DBMS
Spatial DBMS Facilities

The above three topics cover the material in this topic from slightly different perspectives and should be read by anyone working with geometry in tables or drawings stored in a DBMS.

Data Storage Strategies
Database Installations
Database Administrator Edition
Example: Storing a Drawing in Manifold Spatial DBMS
Example: Storing an Image in Manifold Spatial DBMS
Example: Storing a Surface in Manifold Spatial DBMS
Manifold Spatial DBMS Facilities
Linked Drawings
Oracle Spatial Facilities
Project Pane - Open Data Source
Queries and Geoms
Spatial Extensions
SQL Server Spatial DBMS Facilities
Tools - Administrator Console
Tools - Database Console

Manifold Spatial DBMS Facilities
Manifold Spatial DBMS Facilities
Before beginning this topic please read the Spatial DBMS and the Spatial DBMS Facilities topics for an introduction to spatial DBMS usage with Manifold.

In addition to working with a variety of spatial DBMS packages that have built in spatial capability, Manifold can confer upon almost any DBMS the ability to work as a spatial DBMS even if the DBMS vendor has not provided a native geometry type or other built-in spatial infrastructure. Manifold provides such capabilities by providing:

- **Manifold-managed geometry types** - Manifold can operate using either Manifold's own high performance GEOMETRY type or using an open standard such as OGC WKB.
- **Manifold-managed spatial indices** - Manifold can automatically create a spatial index that enables high performance spatial operations.

A spatial index declared by Manifold is processed mostly by the DBMS server and not by the client, so the performance attained will in general be set by the DBMS server, just as with a dedicated spatial DBMS. Editing a linked drawing will automatically update the spatial index.

The main usage of spatial DBMS storage when a GIS package front ends the DBMS is to extract an area of interest so that drawings of manageable size can be viewed and edited in the GIS client even though the DBMS contains a huge, seamless data set that otherwise would be too big to edit in its entirety. Filtering using the
spatial index to achieve extraction of an area of interest subset is likewise done mostly on the server so in this key task using a generic spatial index with GEOMETRY stored in the DBMS works quite well.

Manifold's ability to confer spatial DBMS capabilities to almost any DBMS does not require the use of Manifold's own GEOMETRY type. It could well use generic, non-Manifold, open standards for geometry types such as OGC WKB, and is often deployed in that way. Nonetheless, in this documentation the generic use of Manifold to confer spatial DBMS capability is referred to as Manifold spatial DBMS usage, to indicate it is a capability conferred by Manifold as opposed to being a built-in spatial DBMS capability like that with Oracle Spatial or SQL Server 2008 spatial capabilities.

Requirements

The following are required for Manifold spatial DBMS usage:

- A DBMS that can support binary data types, such as MySQL, Informix, SQL Server 2005, DB2 (not requiring IBM's Spatial Extender), Oracle (used as a regular DBMS exclusive of the built-in spatial features) or PostgreSQL (without spatial extensions).
- Although it is not required when using SQL Server 2005, it is strongly recommended that the Manifold Spatial Extender for SQL Server be installed on SQL Server machines for improved performance. The Manifold spatial extender is a free download.
- Manifold Enterprise Edition or higher edition is required to upload drawings into a DBMS and to confer spatial DBMS capability to that DBMS by creation of a spatial index and supporting facilities. Once such a drawing has been uploaded into the DBMS any Manifold edition can utilize that spatial data set with full spatial DBMS capabilities, including Area of Interest (AOI) windowing.
- Manifold Database Administrator Edition is required to use Administrator Console features such as providing friendly names for components stored in the DBMS. Once such features have been configured using Administrator Console, any Manifold edition can utilize them when connecting to that DBMS.

A Manifold spatial DBMS can store drawings, images and surfaces (which are stored as images).

Exporting and Linking a Drawing

This section uses SQL Server 2005 Express as an example of a generic, non-spatial DBMS. The procedure described will work with almost any modern DBMS.

How to Export a Drawing into SQL Server Express

1. In the project pane right click on the drawing and choose Export.
2. In the Export Drawing dialog choose Data Sources () in the Save as type box.
3. In the Data Source dialog click the data source for the SQL Server Express database you wish to use to highlight it. If a data source for the desired SQL Server Express database has not yet been added to the list of data sources provided in the Data Source dialog, please see the Data Source Dialog topic for instructions on adding a new data source to that dialog. Press OK.
4. In the Export Drawing dialog that pops up choose the fields to be exported with the drawing. By default, all fields except intrinsic fields are checked. Accept the other settings. (Note that the Create spatial index box is checked by default, which will create a spatial index for this drawing and thus confer upon SQL Server Express essential functionality as a spatial DBMS.) Press OK.

The drawing will be exported to SQL Server with a spatial index created.

We can now use Database Console to import or link the drawing into any Manifold project:

1. Launch Database Console and connect to the SQL Server data source.
2. We will see the drawing appear with a Manifold drawing icon next to it. If we have used Administrator Console (available within Database Administrator Edition) to give it a friendly name it will appear using whatever name was given to it in Administrator Console using the Properties dialog.
3. Import or link the drawing by clicking on it to highlight it and then pressing the Import or Link button in the toolbar. Because we declared a spatial index for the drawing we can choose to import or link objects only from a desired area of interest.
Manifold dialogs for export by default use the native Manifold **Geometry** type. Manifold's **Geometry** type is a high performance type that is just as efficient or more so than the native geometry types employed by various spatial DBMS vendors, but if desired some other geometry type such as **Geometry(WKB)** may be used and supported with a spatial index.

The above comments for use of SQL Server 2005 as a spatial DBMS using Manifold geometry and spatial indices apply to almost any big-name DBMS. For example, we can use MySQL in the same way.

**Spatial Indices**

Manifold can establish a spatial index within a DBMS for a database object (a table) that is to be treated as a drawing. A spatial index can only be created for a table that has a primary key. A primary key is created automatically whenever a drawing is exported to a DBMS and whenever the ID field is exported when a table is exported into a DBMS.

Once a spatial index is established for a table that is treated as a drawing, that index will be maintained primarily by the DBMS server. The spatial index allows the DBMS to fetch subsets of objects based upon location and extent, an obvious usage being to fetch manageable subsets of objects for a given area of interest from very large drawings.

A key limitation of using generic spatial indices is that they cannot grow outside the bounding box (the maximum East / West and North / South extents) of the original drawing for which they were created. This makes it impossible to add new objects to the drawing that extend beyond the original, bounding box. There are three ways to get around this limitation when using Manifold-managed spatial DBMS:

- Remove the spatial index, do the desired operation and then re-build the spatial index. Easy to do, but this requires the Administrator Console which is a part of more expensive Manifold editions such as Database Administrator edition and Ultimate Edition.
- Before uploading a drawing into the DBMS, create four temporary objects in the drawing that are at the corners of a larger region than ever will be needed for that drawing. Upload the drawing and the spatial index will be created to fit the larger region. After uploading, link the drawing and delete the temporary objects. This can be done with Enterprise edition, but it does require guessing at what will be the maximum extent required for the drawing.
- Use SQL Server and use the Manifold Spatial Extender for SQL Server. The Spatial Extender runs as a SQL Server add-in and will automatically grow spatial indices as required.

A spatial index declared by Manifold is processed mostly by the DBMS server and not by the client, so the performance attained will in general be set by the DBMS server, just as with a dedicated spatial DBMS. Editing a linked drawing will automatically update the spatial index.

The main usage of spatial DBMS storage when a GIS package front ends the DBMS is to extract an area of interest so that drawings of manageable size can be viewed and edited in the GIS client even though the DBMS contains a huge, seamless data set that otherwise would be too big to edit in its entirety. Filtering using the spatial index to achieve extraction of an area of interest subset is likewise done mostly on the server so in this key task using a generic spatial index with **Geometry** stored in a DBMS works quite well.

**Examples**

The following examples use Microsoft SQL Server Express 2005 as a default example of a DBMS. The examples assume the DBMS has been installed using defaults as set forth in the SQL Server Express Edition topic, that the Manifold spatial extender has been installed on the server machine and that a data source has been created in the **Data Source** dialog to connect to that SQL Server Express database as set forth in the Data Source dialog topic.

**Example: Storing a Drawing in Manifold Spatial DBMS**

**Example: Storing an Image in Manifold Spatial DBMS**

**Example: Storing a Surface in Manifold Spatial DBMS**

**See Also**

**Geometry in Tables**

**Spatial DBMS**
Spatial DBMS Facilities

The above three topics cover the material in this topic from slightly different perspectives and should be read by anyone working with geometry in tables or drawings stored in a DBMS.

- Data Storage Strategies
- Database Installations
- Database Administrator Edition
- Linked Drawings
- Manifold Spatial Extender for SQL Server
- Oracle Spatial Facilities
- Project Pane - Open Data Source

Queries and Geoms

Spatial Extensions

SQL Server Spatial DBMS Facilities

Tools - Administrator Console

Tools - Database Console
Example: Storing a Drawing in Manifold Spatial DBMS

This example uses Microsoft SQL Server Express 2005 as a default example of a non-spatial DBMS to which spatial DBMS capabilities can be added by using Manifold. Such usage is referred to as Manifold Spatial DBMS since Manifold is used to manage the storage, even though the actual geometry stored might be some non-Manifold type such as OGC WKB.

This example assumes the SQL Server Express 2005 DBMS has been installed using defaults as set forth in the SQL Server Express Edition topic and that a data source has been created in the Data Source dialog to connect to that SQL Server Express database as set forth in the Data Source dialog topic.

As advised in the Spatial DBMS topic, users are strongly encouraged to install the Manifold Spatial Extender for SQL Server when using SQL Server 2005 as a spatial DBMS. Procedures using with the spatial extender installed or not are identical, but use of the Manifold spatial extender will improve performance of spatial indices.

Manifold Enterprise Edition is required to export a drawing into a non-spatial DBMS in a way that grants spatial DBMS capabilities. Thereafter, any Manifold edition can utilize those spatial DBMS capabilities for drawings stored using Enterprise Edition. In addition, Manifold Database Administrator Edition provides very useful and convenient features that may be employed by administrators to configure DBMS storage to make life easier for ordinary users using other Manifold editions. Database Administrator Edition is required for the Administrator Console procedures shown later in this example.

Export a Drawing to SQL Server Express 2005

In this example we use Manifold Enterprise Edition to export a sample drawing of Mexico to SQL Server Express 2005:

In the project pane we right click on the drawing and choose Export.

In the Export Drawing dialog we choose Data Sources in the Save as type box.
In the **Data Source** dialog we click on the entry for SQL Server and then press **OK**.

[This screen shot assumes we have previously created an entry for our SQL Server connection as set forth in the Data Sources dialog topic and have named it **Local SQL Server Express**.]

We can see that this particular Manifold installation also has a connection configured to an Oracle Express data source.]
In the Export Drawing dialog we accept defaults and press OK. All fields (columns) will be exported except intrinsic fields. The drawing will be uploaded into the DBMS using standard Manifold geometry types and a spatial index will be created.

**Important:** When exporting a drawing to a spatial DBMS it is best if the drawing does not contain a column called OID. This name is used by Manifold as the default name for the object identity field. If we like, we can still export a drawing that contains such a column but then we will have to remember in the export dialog to change the name of the identity field to something else.

While that is certainly easy enough to do, it is also easy to forget to do in which case the export will fail and we will wonder why. Also, it is useful to always assume that an OID field is the identity field. Therefore, it is strongly recommended to treat this field name as a reserved name.

The Export Drawing dialog gives us some options as to how we store the geometry. The Type box near the top of the dialog in this case will have only one choice, Manifold, unless the DBMS to which we are exporting has native spatial DBMS capabilities. If we are exporting to a non-spatial DBMS such as SQL Server 2005 (as opposed to SQL Server 2008 with the new spatial capability Microsoft has announced for SQL Server 2008) the only choice will be Manifold, meaning that Manifold will provide the spatial DBMS capabilities. If we were exporting to a DBMS with native spatial capabilities, such as Oracle, there would be a choice of the native spatial DBMS type as well, for example, Oracle, in addition to the Manifold choice.
Manifold can support non-Manifold geometry types for drawing storage and such are available within the Geometry type box even though it is a terrible idea for beginners not to use the default choice, Geometry, which chooses the Manifold Geometry type. Manifold geometry is by far the fastest, most robust and most modern of any of the various geometry types available.

In certain rare circumstances experts may prefer to choose Geometry(WKB) to use OGC WKB geometry or even Geometry (SHP) for an ESRI-style, "shapefile"-derived geometry type. If we pick something other than Geometry we are asking for trouble, as the other geometry types don't do as good a job of storing coordinate information as Manifold Geometry. There might be some justification for experts to use WKB if they like OGC ways of doing things and are willing to deal with the difficulties involved, but there is no good reason to use Geometry(SHP) outside of very unusual and exotic applications.

**Link a Drawing from SQL Server Express 2005**

Now that we have a drawing in SQL Server we can import it or link it into Manifold using any Manifold edition. The following sequence, for example, can be used in Manifold Personal Edition or any other Manifold edition.

We open a project and launch Database Console. Within Database Console we choose the SQL Server Express installation from the Data Source dialog if it is not already loaded into the Data Source box. We press the Refresh button to see the contents of the DBMS server.
Database Console shows us the contents of the server. The drawing will appear using a Manifold drawing icon.

By default, Database Console will show us all of the contents of the server. If our budget allows, we can use the Administrator Console feature from Manifold Database Administrator Edition to configure the database to use friendly names so that the display is simplified for new users. See an example below.

To link the drawing into our project we click on it to highlight it and then we press the **Link** button in the Database Console toolbar.
The Import / Link Options dialog allows us to specify a name to use for the linked component and to choose between using all objects or only using those objects which fall within a desired area of interest. Note that area of interest specification, a serious spatial DBMS capability, is available to us even though the DBMS we are using is not a DBMS with built-in spatial features and even though we might not be using Manifold Enterprise Edition. Press OK to continue.

The result is that a new, linked drawing appears in the project.

If we click this drawing open we see it is indeed the sample Mexico drawing exported in the previous part of this example. It has been linked complete with all projection information exactly as it was when exported.

Configure a Drawing using Administrator Console

Being able to export drawings into almost any DBMS with true spatial DBMS capabilities available upon import is extremely useful. However, as seen above the example is limited for two main reasons: First, it exposes complicated DBMS objects and names which might potentially confuse beginning users and second, it does not allow formatting for the linked drawing to be stored on the DBMS server.
We can add both capabilities by using Administrator Console, a feature provided by Manifold Database Administrator Edition and by Ultimate Edition. Once drawings are so configured using Database Administrator edition, thereafter any Manifold edition can connect to the database and use friendly names and formatting stored on the database. For that reason, organizations that have many users using drawings stored within databases will usually have a few Manifold Database Administrator or Ultimate licenses for use by managers to configure drawings within the database to use friendly names and to allow formatting, while the majority of their licenses are Enterprise or Universal licenses.

We will first turn on storage of formatting within the DBMS and then provide a friendly name (component name) to use.

Launch Manifold Database Administrator Edition and connect to the SQL Server Express database with Administrator Console. Press Refresh to see the contents of the database.

We note that the column showing format status is not turned on in the display. To turn it on, we press the Columns button.
In the **Columns** dialog we check the **Format** box to display that column.

We double-click into the **Format** cell for the entry for the Mexico drawing and set the **Format** value to **Yes**.

Note that Administrator Console shows all database objects using their database names. It's easy to tell which are drawings because they will have a projection and a spatial index value.

Next we will provide a component name, that is, a friendly name.
To do so, double-click into the Component cell for the Mexico drawing and provide a desired name. For lack of anything better coming to mind, we will use the name Mexico. This name will be displayed when Database Console shows the contents of the database.

**Linking a Drawing Configured with Administrator Console**

Now that the drawing in the database has been configured with Administrator Console we can use any Manifold edition to import it or link it, with the benefits of friendly names and formatting on the database.

Launch any Manifold System edition, open a project and launch Database Console. Connect to the SQL Server Express database used and press the Refresh button.

Database Console will see that component names are available in the database and will automatically engage the Component View button to display database items in the console view using friendly names. Instead of a list of confusing database objects we see Manifold components using friendly names as configured by Administrator Console in the previous parts of this example. Excellent!

To link a drawing, click on it to highlight it and then press the Link button.
When we open the drawing we can apply formatting, and the formatting will be stored on the database so that any other user linking the drawing will see that formatting.

Notes

The procedures used herein will work with almost any DBMS that can store binary data, such as MySQL, Informix or others. SQL Server Express 2005 is used as a convenient example since many people have it already. Oracle Express or DB2 could have been used just as conveniently. However, since Oracle Express includes built-in Locator capability (Oracle's spatial DBMS technology for drawings as in Oracle Spatial) and DB2 Express-C can be equipped with the free DB2 Spatial Extender, we have used SQL Server 2005 to provide an example of a modern DBMS that does not include built-in spatial DBMS.

When discussing spatial storage in SQL Server one should keep in mind there are several possibilities depending upon which version of SQL Server is being used and which spatial technology is employed. See the discussion in the Spatial DBMS topic.

See Also

Geometry in Tables
Manifold Spatial DBMS Facilities
Spatial DBMS
Spatial DBMS Facilities

Example: Storing an Image in Manifold Spatial DBMS
Example: Storing a Surface in Manifold Spatial DBMS

Data Storage Strategies
Database Installations
Database Administrator Edition
Linked Drawings
Manifold Spatial Extender for SQL Server
Oracle Spatial Facilities
Project Pane - Open Data Source
Queries and Geoms
Spatial Extensions
SQL Server Spatial DBMS Facilities
Tools - Administrator Console
Tools - Database Console
Example: Storing an Image in Manifold Spatial DBMS

This example uses Microsoft SQL Server Express 2005 as a default example of a non-spatial DBMS to which spatial DBMS capabilities can be added by using Manifold. Such usage is referred to as Manifold Spatial DBMS since Manifold is used to manage the storage.

This example assumes the SQL Server Express 2005 DBMS has been installed using defaults as set forth in the SQL Server Express Edition topic and that a data source has been created in the Data Source dialog to connect to that SQL Server Express database as set forth in the Data Source dialog topic.

As advised in the Spatial DBMS topic, users are strongly encouraged to install the Manifold Spatial Extender for SQL Server when using SQL Server 2005 as a spatial DBMS. Procedures using with the spatial extender installed or not are identical, but use of the Manifold spatial extender will improve performance of spatial indices.

Manifold Enterprise Edition is required to export an image into a non-spatial DBMS in a way that grants spatial DBMS capabilities. Thereafter, any Manifold edition can utilize those spatial DBMS capabilities for images stored using Enterprise Edition. In addition, Manifold Database Administrator Edition provides very useful and convenient features that may be employed by administrators to configure DBMS storage to make life easier for ordinary users using other Manifold editions. Database Administrator Edition is required for the Administrator Console procedures shown later in this example.

Export an Image to SQL Server Express 2005

In this example we use Manifold Enterprise Edition to export a sample image to SQL Server Express 2005:

We open a project with the image.

If we open the image we see it shows portion of the Microsoft campus in Redmond, Washington.
This image is reasonably well detailed, at about 0.3 meter resolution, as we can see by zooming in.

To export the image to a database we right click on the image in the project pane and choose Export.

In the Export Image dialog we choose Data Sources in the Save as type box.
In the **Data Source** dialog we click on the entry for SQL Server and then press **OK**.

This screen shot assumes we have previously created an entry for our SQL Server connection as set forth in the Data Sources dialog topic and have named it **Local SQL Server Express**.

We can see that this particular Manifold installation also has a connection configured to an Oracle Express data source.

In the **Export Image** dialog we accept defaults and press **OK**. The image will be uploaded into the DBMS using standard Manifold spatial DBMS surface storage technology for raster data sets (images and surfaces). Manifold stores images as tiles so that subsections of images can be retrieved very rapidly. The default **Tile size** recommended by Manifold may be adjusted up or down, but usually is the best choice.

The option to **Create Pyramids** creates pre-computed, intermediate views, called **pyramids**, which enable faster display of very large images when zoomed out. **Compression** using the default **Deflate** option is a good
idea because the computational cost of decompression is negligible and there is no image quality lost, yet having a smaller amount of data in the database not only saves disk space but it helps the DBMS perform faster.

**Note:** Export of images to databases is robust. For example, exporting an image to a database as a tiled table (the actual storage mechanism used within the database) correctly rolls back all changes upon a failure.

**Link an image from SQL Server Express 2005**

Now that we have an image in SQL Server we can import it or link it into Manifold using any Manifold edition. The following sequence, for example, can be used in Manifold Personal Edition or any other Manifold edition.

We open a project and launch Database Console. Within Database Console we choose the SQL Server Express installation from the Data Source dialog if it is not already loaded into the Data Source box. We press the **Refresh** button to see the contents of the DBMS server.
Database Console shows us the contents of the server.

**Important:** In this example, we have continued using the same SQL Server Express installation employed in the Example: Storing a Drawing in Manifold Spatial DBMS topic. At the end of that example we used Administrator Console to configure a friendly name for the drawing that was stored. When Database Console connects to a DBMS that contains friendly names it automatically turns on with the Component View button enabled, so that only friendly names are shown and database object names that might be confusing to inexpert users are hidden.

Uploading an image into a DBMS uploads it without a friendly name until we configure a friendly name using Administrator Console. To see the image that was just uploaded we will have to turn Component View off to see everything in the database.
The image will appear using a Manifold image icon. To link the image into our project we click on it to highlight it and then we press the **Link** button in the Database Console toolbar.

Note that unlike the case of importing or linking a drawing as seen in the previous Example, no **Import / Link Options** dialog appears when importing or linking an image to allow specifying a subset of the image for the area of interest. The reason is that the nature of image storage technology using tiles is such that at any moment there is, in effect, a virtual area of interest subset acquisition from the database, with tiles being fetched as needed for whatever view is commanded through panning and zooming a display window.

If we click this image open we see it is indeed the image exported in the previous part of this example. It has been linked complete with all projection information exactly as it was when exported.

**Configure an image using Administrator Console**

Being able to export images into almost any DBMS with true spatial DBMS tiled storage is extremely useful. However, as seen above the example is limited because it exposes complicated DBMS objects and names which might potentially confuse beginning users.

We can avoid complicated database names by using friendly names (component names) as provided through Administrator Console, a feature that is part of Manifold Database Administrator Edition and by Ultimate Edition. Once images are so configured using Database Administrator edition, thereafter any Manifold edition can connect to the database and use friendly names. For that reason, organizations that have many users using images stored within databases will usually have a few Manifold Database Administrator or Ultimate licenses for use by managers to configure images within the database to use friendly names, while the majority of their licenses are Enterprise or Universal licenses.

Using Database Administrator Edition we will specify a friendly name (**component name**) to use. Launch Manifold Database Administrator Edition and connect to the SQL Server Express database with Administrator Console. Press **Refresh** to see the contents of the database.
Administrator Console shows all database objects using their database names. It's easy to tell which are images because they will have a Manifold image icon.

To provide a component name, that is, a friendly name, double-click into the Component cell for the Microsoft image and provide a desired name. For lack of anything better coming to mind, we will use the name Microsoft HQ. This name will be displayed when Database Console shows the contents of the database.

**Linking an image Configured with Administrator Console**

Now that the image in the database has been configured with Administrator Console we can use any Manifold edition to import it or link it with the benefit of friendly names.
Launch any Manifold System edition, open a project and launch Database Console. Connect to the SQL Server Express database used and press the **Refresh** button.

Database Console will see that component names are available in the database and will automatically engage the Component View button to display database items in the console view using friendly names. Instead of a list of confusing database objects we see Manifold components using friendly names as configured by Administrator Console in the previous parts of this example.

To link an image, click on it to highlight it and then press the **Link** button.

![Microsoft HQ](image)

The result is an image within the project pane that is identical to the original image we exported.

**Notes**

New users may wonder about the need to switch off **Component View** to see the newly-uploaded image before a friendly name has been specified with Administrator Console. Would that not be an inconvenience to most users?

As a practical matter, no. When Manifold users have Administrator Console available it is used to assign friendly names when drawings, images or surfaces are uploaded into a database. In that case, the **Component View** button will always be on and users will always see the contents of their databases using friendly names. Friendly names are such a convenience that most users, even experts, strongly prefer them to raw, database names for items within a database.

In contrast, if a Database Administrator Edition license has not been acquired then no friendly names will be in use, the **Component View** button will always be off and users will always see the contents of their databases using the naming style used by that database. Many users will not mind that if their budgets do not allow for a Database Administrator Edition license.

See also the notes on SQL Server Express as an example DBMS at the end of the previous Example topic.
This example uses SQL Server 2005. We can also use a spatial DBMS such as SQL Server 2008 to store images using Manifold spatial capabilities. See the Example: Storing an Image in SQL Server 2008 topic for such an example.

**See Also**

Geometry in Tables
Manifold Spatial DBMS Facilities
Spatial DBMS
Spatial DBMS Facilities

Example: Storing a Drawing in Manifold Spatial DBMS
Example: Storing a Surface in Manifold Spatial DBMS
Example: Storing an Image in SQL Server 2008

Data Storage Strategies
Database Installations
Database Administrator Edition
Linked Drawings
Manifold Spatial Extender for SQL Server
Oracle Spatial Facilities
Project Pane - Open Data Source
Queries and Geoms
Spatial Extensions
SQL Server Spatial DBMS Facilities
Tools - Administrator Console
Tools - Database Console
Example: Storing a Surface in Manifold Spatial DBMS

This example uses Microsoft SQL Server Express 2005 as a default example of a non-spatial DBMS to which spatial DBMS capabilities can be added by using Manifold. Such usage is referred to as Manifold Spatial DBMS since Manifold is used to manage the storage.

This example assumes the SQL Server Express 2005 DBMS has been installed using defaults as set forth in the SQL Server Express Edition topic and that a data source has been created in the Data Source dialog to connect to that SQL Server Express database as set forth in the Data Source dialog topic.

As advised in the Spatial DBMS topic, users are strongly encouraged to install the Manifold Spatial Extender for SQL Server when using SQL Server 2005 as a spatial DBMS. Procedures using with the spatial extender installed or not are identical, but use of the Manifold spatial extender will improve performance of spatial indices.

Manifold Enterprise Edition is required to export a surface into a non-spatial DBMS in a way that grants spatial DBMS capabilities. Thereafter, any Manifold edition can utilize those spatial DBMS capabilities for surfaces stored using Enterprise Edition. In addition, Manifold Database Administrator Edition provides very useful and convenient features that may be employed by administrators to configure DBMS storage to make life easier for ordinary users using other Manifold editions.

Exporting a surface is essentially identical to exporting an image as shown in the Example: Storing an Image in Manifold Spatial DBMS topic in that surfaces are stored within databases as though they were images, but with a Height channel. The difference between surfaces and images arises upon import, when the Height channel is imported to recreate the surface. At the present time, surfaces cannot be linked from a database.

Export a surface to SQL Server Express 2005

In this example we use Manifold Enterprise Edition to export a sample surface to SQL Server Express 2005:

We open a project with the surface to be exported.
If we open the surface we see it shows the Montara Mountain sample elevation data set. The surface has been colored with a palette using the View - Display Options dialog.

To export the surface to a database we right click on the surface in the project pane and choose Export.

In the Export Surface dialog we choose Data Sources in the Save as type box.

In the Data Source dialog we click on the entry for SQL Server and then press OK.

[This screen shot assumes we have previously created an entry for our SQL Server connection as set forth in the Data Sources dialog topic and have named it Local SQL Server Express.

We can see that this particular Manifold installation also has a connection configured to an Oracle Express data source.]
In the Export Surface dialog we accept defaults and press OK. The surface will be uploaded into the DBMS using standard Manifold spatial DBMS surface storage technology for raster data sets (images and surfaces). As with images, Manifold stores surfaces as tiles so that subsections of surfaces can be retrieved very rapidly. The default Tile size recommended by Manifold may be adjusted up or down, but usually is the best choice.

The option to Create Pyramids creates pre-computed, intermediate views, called pyramids, which enable faster display of very large surfaces when zoomed out. Compression using the default Deflate option is a good idea because the computational cost of decompression is negligible and there is no surface quality lost, yet having a smaller amount of data in the database not only saves disk space but it helps the DBMS perform faster.

Import a surface from SQL Server Express 2005

Now that we have a surface in SQL Server we can import it into Manifold using any Manifold edition. The following sequence, for example, can be used in Manifold Personal Edition or any other Manifold edition.

We open a project and launch Database Console. Within Database Console we choose the SQL Server Express installation from the Data Source dialog if it is not already loaded into the Data Source box. We press the Refresh button to see the contents of the DBMS server.

Important: In this example, we have continued using the same SQL Server Express installation employed in the Example: Storing a Drawing in Manifold Spatial DBMS topic and in the Example: Storing an Image in Manifold Spatial DBMS topic. At the end of those examples we used Administrator Console to configure a friendly name for the drawing or the image that was stored. When Database Console connects to a DBMS that contains friendly names it automatically turns on with the Component View button enabled, so that only friendly names are shown and database object names that might be confusing to inexpert users are hidden.
Uploading a surface into a DBMS uploads it without a friendly name until we configure a friendly name using the Administrator Console. To see the surface that was just uploaded we will have to turn Component View off to see everything in the database.

The surface will appear using a Manifold image icon, because it exists within the DBMS as a raster data set, the same as images. To import the surface into our project we must first click on the + icon to the left of the image icon to expand the entry so that the different channels of the image can be seen. We can then click on the Heights channel to highlight it and then we press the Import button in the Database Console toolbar.
The result is that a new surface appears in the project.

Note that unlike the case of importing or linking a drawing as seen in the Example of storing drawings to a DBMS, no Import / Link Options dialog appears when importing a surface to allow specifying a subset of the surface for the area of interest.

If we click this surface open we see it is indeed the surface exported in the previous part of this example, albeit with no formatting specified. It has been imported complete with all projection information as it was when exported.

Notes

If we have Database Administrator Edition we can configure surfaces stored in databases with friendly names just as we can with drawings and images. See the Example: Storing an Image in Manifold Spatial DBMS topic for an example of configuring friendly names.

New users may wonder about the need to switch off Component View to see the newly-uploaded surface before a friendly name has been specified with Administrator Console. Would that not be an inconvenience to most users?

As a practical matter, no. When Manifold users have Administrator Console available it is used to assign friendly names when drawings, surfaces or surfaces are uploaded into a database. In that case, the Component View button will always be on and users will always see the contents of their databases using friendly names. Friendly names are such a convenience that most users, even experts, strongly prefer them to raw, database names for items within a database.

In contrast, if a Database Administrator Edition license has not been acquired then no friendly names will be in use, the Component View button will always be off and users will always see the contents of their databases
using the naming style used by that database. Many users will not mind that if their budgets do not allow for a Database Administrator Edition license.

See also the notes on SQL Server Express as an example DBMS at the end of the Example topic on exporting drawings.

See Also

Geometry in Tables
Manifold Spatial DBMS Facilities
Spatial DBMS
Spatial DBMS Facilities

Example: Storing a Drawing in Manifold Spatial DBMS
Example: Storing an Image in Manifold Spatial DBMS

Data Storage Strategies
Database Installations
Database Administrator Edition
Linked Drawings
Manifold Spatial Extender for SQL Server
Oracle Spatial Facilities
Project Pane - Open Data Source
Queries and Geoms
Spatial Extensions
SQL Server Spatial DBMS Facilities
Tools - Administrator Console
Tools - Database Console
Programming

Programming Manifold

Manifold may be programmed in several ways:

- "Ad hoc" programming using Active Columns in tables or ViewBots.
- Using SQL queries in the classic DBMS way to alter table structure and otherwise make updates.
- Using Spatial SQL for sophisticated spatial operations on vector (drawings) or raster (images and surfaces) data or a combination thereof.
- Writing scripts using ActiveX scripting languages such as VBScript (the scripting edition of Visual Basic), JScript (also called JavaScript), Perl or Python, or .NET languages such as C#, VB .NET or JScript .NET. Scripts components may be created for Forms or may be created as standalone Script components.
- Scripts may invoke and control user interface elements such as dialogs. See the User Interface Scripting topic.
- Calling Manifold System from external programs written in standard languages like Visual Basic and Visual C++ or their .NET language equivalents, such as Visual Basic .NET, C#, Iron Python, etc.
- Adding scripts (which may call external COM objects) as custom toolbar buttons or menu items. See the Add-Ins topic for information on adding new toolbar buttons or menu commands.
- Programming in ASP or ASP.NET pages to create custom web pages that utilize Manifold IMS to create dynamic, Manifold-enabled web sites. Web applications on intranets are quite often the fastest, easiest and lowest cost way of creating customized GIS applications for distribution and use within an organization.

There is no additional programming language or toolkit that must be purchased in order to program Manifold. Every system includes the ability to use ActiveX languages or .NET languages to write sophisticated, forms-based programs using standard Windows controls. If you do have an external programming environment such as Visual Studio .NET, you can use that as well without the need to purchase any other options. You can even get a free Express version of Visual Studio from Microsoft.

Manifold also provides a Debugger for ActiveX scripting similar to those used in professional programming environments. The Debugger allows stepping through multiple scripts with automatic stops at breakpoints, paused execution and stepping into, over and out of routines. The Call Stack, Variables and Watches panes provide dynamic, editable displays of variables, functions, computed values and contexts. The Debugger works with VBScript, JScript and any other ActiveX scripting engine that supports Microsoft debugging calls. It is not used when scripting in a .NET language like C#.

Advice for Applications Developers

Many developers are creating commercial products based upon Manifold. There are two approaches, standalone applications or web applications, which in some sophisticated applications may be combined:

Standalone Applications

In this traditional way of creating and distributing applications a developer acquires however many Manifold licenses are required for the development team and an application is created that utilizes Manifold features through the API set forth in the Programming Reference chapter.

The application is then deployed using Manifold runtime licenses, which include the full Manifold API but not the Manifold GUI. A Manifold runtime license must be installed on every machine on which an application that calls Manifold facilities is deployed. That will require some means of keeping track of the serial number for each license and arranging for activation of each runtime license. Different developers have different preferences on how to arrange for installation and activation of the Manifold runtime license.

At one end of the spectrum are developers who tell their customers to acquire, install and activate a Manifold runtime license before installing the developer’s product. In this scenario the developer leaves it up to the customer to buy and to install and activate the necessary Manifold runtime. There are advantages and disadvantages to this strategy.

On the plus side, this strategy reduces the developer’s use of cash to acquire and inventory Manifold runtime licenses and it eliminates the need for the developer to figure out how to acquire a Manifold license and then
remotely install and activate it for the customer. However, on the minus side this strategy does risk that a potentially inexperienced customer might not license the right Manifold product or install and activate it correctly. It also makes it clearly known to the customer that Manifold is the engine used for the application. A final disadvantage is that if the developer does not copy protect the application then he or she will soon find out that a significant percentage of customers buy one application but then will acquire many Manifold runtime licenses to enable installation of that one application on many machines.

Many developers will charge much higher prices for their application than the price of a Manifold runtime license and so they will not want it advertised that their application is based upon Manifold. Such developers will prefer to handle all parts of the installation and will install the Manifold runtime as part of their application's installation. There are many different approaches to managing serial numbers and activating the runtime installation.

One approach favored by developers who send personnel to install applications and train customers is to have the developer's own personnel install and activate the runtime. A second approach favored by developers with applications that are intended to be installed by the customer is to use an automated approach to Manifold activation.

The usual automated approach is that the developer creates a small, web-hosted DBMS application that contains the serial numbers acquired by the developer for the Manifold runtime licenses acquired by the developer. When the developer sells an application he or she assigns the new customer a hash code that corresponds to an available license. The developer's application installation script includes a dialog in which the customer provides the hash code, allowing the installation script to connect to the developer's DBMS, grab the serial number associated with that hash code and then activate the Manifold runtime (using command line activation) with that serial number.

Either technique allows the developer to utilize Manifold's serial number and Activation key copy protection mechanism as a handy way of providing copy protection for their own applications as well (since the developed application won't run without an authorized Manifold runtime license). Developers are sometimes annoyed with the need to deal with Manifold activation until they realize how that mechanism discourages piracy of their own applications, at which point they become enthusiastic supporters of that Manifold mechanism.

See the Activation Keys and Serial Numbers topic for a discussion of the Manifold serial number and activation key mechanism. See the Developing Applications topic for specific advice of interest when developing standalone applications.

Web Applications

When customers are known to have an Internet connection available, many lightweight applications can be implemented as web applications that utilize Manifold IMS on a web server and free web browsers on client machines. In this case a developer will acquire Manifold licenses as needed for the development team and then deploy the application to a web server using either a runtime license or a regular license. Developers will often deploy to a web server using a full license because costs are so low in any case that they deem it worthwhile to have a fully interactive Manifold GUI available on the web server should it ever be necessary for debugging purposes.

There are many advantages to such an architecture, the most obvious being that there is near-zero cost for Manifold since client-side software consists of a free browser and the server-side software could consist of only a single Manifold runtime license. There are other advantages to web applications as well:

- **Control over data** - The applications developer can retain full control over data because it is resident on the server and need not be installed on client machines.
- **Easier data updates** - Changing data on the server instantly updates all client usage without need to distributed updated files to client machines.
- **Easier application updates** - Updating the application on the server updates the application for all clients.
- **Potential for subscription model** - A web based application allows business models that charge for application usage in real time. This allows conversion of what is usually a one-time purchase into a subscription or annuity, which can assist cash flow for developers.
- **No redistribution issues** - Because no vendor software is used client-side there is no need for the applications developer to deal with redistributing or activating runtime licenses into client machines.

Of course there are disadvantages to web applications as well. Web applications are limited by Internet bandwidth to lower performance applications than are possible with standalone applications. See the GIS and Networking and the What about Ajax? topics for a discussion of the issues involved.
Web applications can also require a greater amount of server-side programming in comparison to the local programming many programmers are familiar with using tools like the Microsoft Express series.

**Advice for New Programmers**

Because Microsoft languages and language standards are used within Manifold, to program Manifold we need to understand the Microsoft approach to languages and to Windows programming. In addition, although Manifold System is itself coded entirely in Microsoft Visual C++, many applications developers prefer to work with Microsoft's Visual Basic languages for customization of applications.

Experienced programmers already have strong opinions as to the languages they prefer. In modern times, most experienced programmers are already working with Visual Studio .NET using some .NET language they like. In contrast, beginners often start with scripts using some scripting language and, in the Microsoft world, often begin using some version of Visual Basic. It is therefore important to understand the differences between various versions of Visual Basic offered by Microsoft. This topic provides some guidance for beginners, to help them avoid traps such as coding in VBA.

**Microsoft BASIC and Scripting Languages**

At the present time, there are several Microsoft languages of interest to beginning Windows developers who do scripting or who prefer to work with a BASIC language:

- **Javascript (JScript or JScript .NET)** - JScript is Microsoft's superb implementation of the widely used Javascript language, a scripting version of the Java language. Javascript is a favorite of "cross platform" writers creating Internet applications who wish to write scripts that work with many different browsers on Windows and non-Windows operating systems. JScript is an interpreted language that executes within an ActiveX scripting engine.

- **Visual Basic (VB or VB .NET)** - A full-fledged development language that is normally used within Microsoft's Visual Studio development environment. Visual Basic is a compiled language. Many professional applications that are not coded in Visual C++ are coded in Visual Basic. Visual Basic includes superb editing, forms creation and other accessories that facilitate use of this language.

- **Visual Basic Scripting Edition (VBScript)** - A subset of Visual Basic that is a very widely used scripting language, in part because it comes free with every copy of Microsoft's Internet Explorer browser and may be freely redistributed with applications created using Visual Studio. VBScript is the scripting language of choice (with JScript being a second choice) for Microsoft web applications. It is the main language used with ActiveX scripting. Microsoft's push into Internet has made VBScript immensely popular and well supported by third parties with numerous books and objects. An interpreted language, VBScript has become so popular that third parties (such as manifold.net) are now extending it with forms creation and other tools. VBScript is virtually identical to VB so expertise in either VB or VBScript is automatically useful in the other language as well. VBScript has also emerged as the main language used to configure and customize Microsoft's Enterprise class servers.

- **Visual Basic for Applications (VBA)** - A VB variation that likely would have no life were it not bundled with Microsoft's own Office applications. This is a Visual Basic of sorts that is embedded within an application. It is an interpreted language, so any programs or add-ins written in it automatically deliver the source code for the add-in to every user (a real showstopper for most commercial developers). In recent years Microsoft has tried to sell VBA under license for use by third parties; however, despite a marketing push by Microsoft as of this writing fewer than 200 non-Microsoft applications use VBA. There are two main reasons VBA has not caught on: For professional applications, VBA has a high hassle-factor without offering the benefits of true Visual Basic or Visual Studio. For casual use VBA is not significantly better than VBScript while costing far more (potentially adding hundreds of dollars of end user cost) and being clumsier. At one point, VBA had a richer set of tools than VBScript but the surging popularity of VBScript has erased that advantage.

Even within Office application programming, VBA is being pushed aside by programming in real VB or the use of VBScript. As popular as VBA has been for amateur programmers in the Office application programming niche, the engineering team at manifold.net believe that VBA is a technological and business dead-end. We think that the highest value approach to programming with applications in Microsoft operating systems is a two-fold approach using VBScript and VB:

- Use Visual Basic Scripting or JScript for rapid scripting within Manifold. There is so much support for these languages as a result of the growth of Internet that they are clearly the way to go for casual scripting. By adding some elegant support facilities such as Manifold's form designer, they can be used to do everything VBA can do, but without the overhead, cost, hassle-factor and weirdness of VBA.
Scripts written in VBScript or JScript are perfect for use within Active Columns as well. If possible, move to the .NET versions of these languages in order to take advantage of .NET capabilities.

- **Use Visual Basic or Visual C++ within Visual Studio** for extensive applications development with Manifold. This allows source code protection within compiled code, the speed advantages of compiled code and most importantly the ability to use the full array of professional code-warrior tools available within Studio. SourceSafe alone is worth the jump to Studio when working in teams and the Visual Studio editor is a dream come true.

- Of course, when coding in .NET environments (a good idea), use Visual Basic .NET or JScript .NET or C#, the .NET equivalents of the above two strategies. Although coding with .NET involves a steeper learning curve for the complete beginner, almost all professional programmers working in Microsoft environments are writing .NET code. If writing code is an important part of your professional life, the sooner you get a copy of Visual Studio .NET and get your head around .NET, the better.

The above, two-fold approach appears to be the strategy chosen by Microsoft itself for all of its next generation products. A review of Microsoft's online Knowledge Base and other resources on the www.microsoft.com site shows extensive Microsoft use of VBScript or Visual Studio tools for .NET and the Enterprise servers such as SQL Server, Commerce Server and Exchange Server, but virtually no new work in VBA outside of a handful of consumer examples for Office.

Users of Visual Studio need no encouragement for coding in this superb environment. For world-class, professional program development, get a copy of Visual Studio .NET and learn to code in .NET using Visual C++ or Visual Basic .NET. Microsoft makes it easy for people to get started by providing free downloads of the Express version of Visual Studio. Go for it!

To support the use of VBScript for casual scripting, manifold.net has developed forms-based, drag and drop tools drawn from our experience with Windows and Visual Studio that make it very easy to write scripts of amazing sophistication and power in VBScript. In fact, VBScript is so good and the Manifold development environment so easy to use, it is quite likely that professional developers as well as proficient amateurs might reach first for VBScript even for complex tasks.

Using VBScript also means that no additional software is required to program scripts of great sophistication in Manifold. There is no need to purchase Visual Basic or other Visual Studio components, no need to buy any third party accessories, no SDK, and no embedded license fee for VBA. If you have Manifold System you can program right away using true Windows, event-driven and forms-based programming methods. VBScript is a fine "first language" to learn for someone who has never programmed before. It's fast, easy, fun and when it is used with ActiveX controls it is extremely powerful for many practical purposes.

VBScript scripts in Manifold can call any standard ActiveX object if desired. For example, one can add a third party ActiveX control to a form that draws a custom picture of some sensing instrument's readout. There are many thousands of freeware, shareware and inexpensive ActiveX controls we can use in Manifold scripts. Manifold provides a large "starter set" of fun controls like calendars, toolbars and other useful items.

Although Manifold scripts may be written in VBScript or JScript (or even PERL or Python) all of the documentation examples and other support resources from Manifold are based on VBScript. Technically, any ActiveX scripting engine language that you have installed in your machine can be made to work. However, to keep documentation and tech support down to a realistically manageable size we focus on VBScript. There is no technical support available from manifold.net for scripting languages other than VBScript. If you choose to work in other languages it is up to you to learn enough about VBScript and Microsoft methods to figure out how to translate what is said in a VBScript context to your language.

This documentation also assumes the programmer is fluent in Windows programming. Windows programming is based on an **event-driven** model, which is very different than traditional, "procedural" applications programming. Although there are tips scattered throughout this documentation (including the brief introduction below), if you are not fluent in Windows programming you will need to get some good books on Visual Basic, VBScript, Windows, ActiveX and other key Microsoft products and concepts. There are hundreds of good choices. Because Manifold uses standard Microsoft languages and methods you have an entire world of resources to utilize.

One small caveat: because VBScript is so popular for web stuff scripting, many books will discuss it in the middle of lots of other talk about HTML, Active Server Pages (ASP) and other web topics. That's OK. The same characteristics that make VBScript a good first choice in web programming also make it ideal for casual scripting in Manifold as well. The only differences between VBScript books with a web orientation and VBScript use in Manifold will be the environment within which one actually writes the scripts.

**Windows Programming**
Almost everything we see in a Windows application is itself a window. That includes not only the usual "windows" in which applications such as Windows Explorer or Manifold run, but also all the various command buttons, toolbar buttons, dialogs, menu bars and other items one sees on screen. A "window" is just a rectangular region. Each window in Microsoft Windows is assigned a unique id number called the window handle or hWnd.

The operating system continuously surveys each window for any activity. Different types of activity, such as a mouse click or the window being uncovered by another window is an event. When the operating system sees an event occur, it sends a message to all windows, broadcasting news of the event to literally all of the windows. Each window then decides what to do with that message by consulting whatever instructions it has on hand for handling messages of that sort. For example, a window can repaint itself if it has been exposed by another window. Most messages are handled for us automatically by the Windows operating system so we rarely need to think about low-level housekeeping details. Higher level messages are exposed as event procedures that we can utilize as we see fit.

Event-Driven Programming

Traditional, "procedural" languages like the original BASIC language are conceptually quite simple: the program begins at the first line of code and then steps through all the other lines of code in order, making departures from time to time to visit procedural subroutines or to follow loops.

Event-driven programming is quite different, since any part of the code might be called at any time when an event that triggers it occurs. We have to write our code so that any assumptions we make we know will be true at all times. For example, we should keep an "OK" command button disabled if it requires user entry of text into an edit box until we know the edit box actually contains text. Since programs as well as humans can write text into a text box (and thus trigger an event), we have to keep in mind that things may change without human user intervention. This model requires some more thought but it is wonderfully flexible and often results in far more reusable and adaptable code than traditional procedural code.

Forms-Based Programming

Traditional languages involve development along a typical path consisting of three steps: writing the code, compiling it, and then running it. We iterate through this path over and over when creating an application. Creating the user interface in traditional applications is an interwoven part of writing the code and at times will result in spaghetti code connections between various parts of the program to service different user interface situations.

Because everything in Windows programming revolves around windows, events and messages, we take a different approach in most Windows programming and use Forms-Based programming. This involves three steps:

- Create the user interface (a form).
- Set properties for the controls used in the form.
- Write code within a script that implements program logic and tells the controls what to do.

Forms are used to create the windows used to interface to your script or application. They are also used to view tables and otherwise interact with databases through a forms-based human interface. In Manifold, we use the File - Create - Form menu command to create a new form. We can then pop open that form and add controls to it from the Tools toolbar, from the Tools (Advanced) toolbar or from other sources.

Next, we set the properties for each item in the form if we would like to change them from the defaults. This lets us take advantage of the very rich set of automated user interface capabilities Windows provides to control appearance and behavior and otherwise deal with events.

For example, instead of showing a command button as a simple 3D button with a caption we can show it as button with an image on it. We can even use different images in the up and down position, if desired. If we like, in about three seconds we can change the button's properties so that whenever the mouse cursor passes over it the cursor will change to a custom graphic. Cool!

Finally, we open the script associated with the form and write the code that powers the form in VBScript using classic ActiveX, Visual Basic and Windows ideas such as objects, object properties, and object methods. The discipline of having a form that keeps our controls and their properties well organized is a great help in keeping our code clearly organized.
The above sequence of three steps in forms-based programming is typical of Visual Basic application development within Visual Studio. The magic of Manifold System is that the above methodology also may be used with VBScript within Manifold scripts as well. This allows rapid development of sophisticated scripts that provide first-rate user interfaces created in a point-and-click manner.

**ActiveX Languages**

Scripting within Manifold is implemented using either .NET languages or ActiveX scripting. This means that you can use any language for scripting within Manifold for which an ActiveX scripting engine is installed on your computer. For the present, VBScript and JScript are installed with Internet Explorer, a required prerequisite for running Manifold.

At the present writing, downloadable, free-of-charge ActiveX scripting engines for PERL and Python are available from sources such as [www.activestate.com](http://www.activestate.com). End users may download these engines, install them and then merrily write Manifold scripts in either PERL or Python. Manifold.net personnel have experimentally used PERL and Python enthusiastically with both languages on a casual basis.

Other scripting languages, such as REXX and Tcl/Tk, are said to be available as ActiveX scripting engines and presumably could be used within Manifold. These have not been used at Manifold.net but we would be interested in hearing from users who have experimented with them or with any other scripting languages within Manifold.

The practical limitations of development and technical support mean that Manifold's form based programming facilities are designed for VBScript. Users of other languages should not expect to be able to use Manifold's forms and drag and drop controls with those languages. If you work with those other languages you should plan on scripting within the ordinary, text-based facilities of that language. Nonetheless scripts of immense power can be created using languages other than VBScript. We think it's neat that Manifold users can choose the scripting language they like best for the task at hand.

**Choosing a Scripting Language**

To specify a language other than VBScript, open a script for editing in a script window and then choose Script - Language. This dialog allows choice of which scripting language is used. Any scripting language installed in your Windows system may be used. Examples in this documentation are provided for VBScript, JScript, ActivePython, and ActivePERL (the latter two languages from [www.activestate.com](http://www.activestate.com)). Overall, we suggest use of VBScript, JScript, C#, JScript .NET or Visual Basic .NET.

Although .NET languages like C# can have a slightly steeper learning curve for total beginners than ActiveX scripting languages like VBScript or JScript, it is clear that more and more people, even beginners, are writing their Manifold scripts in .NET languages.

**A Future for VBA Enthusiasts**

Avid VBA users may rise to the defense of VBA and may complain this topic overstates the case against VBA. It's true that VBA is a beloved first introduction to VB for many people who are transitioning from casual macro creation and application configuration into serious, professional programming. For that, one should be grateful for VBA, but one's gratitude should not cloud one's eyes into not seeing the many disadvantages of VBA compared to real VB, VB .NET or VBScript.

If you have spent years working with VBA, be happy that you have invested time and effort into developing expertise that is readily portable into VB, VB .NET and VBScript. If you like VBA you'll love working in VB .NET in Visual Studio .NET and you'll feel right at home snapping out sparkling web stuff and Manifold scripts in VBScript.

**.NET and Version Skew**

Manifold utilizes Microsoft's .NET Framework which is being actively evolved for everyone's benefit by Microsoft. Developers should therefore take care to pay attention to what versions of the .NET Framework and their development tools are in use.

For example, Visual Studio .NET 2003 only supports .NET 1.1 assemblies, not .NET 2.0 assemblies. Manifold 6.50 used .NET 1.1 and so could be used with Visual Studio .NET 2003. In contrast, however, Manifold 6.50 SP1 uses .NET 2.0 and so requires Visual Studio .NET 2005 or later.
If we do not yet have Visual Studio .NET 2005, we can download various Express editions of it for free from the Microsoft web site.

**Scripting the User Interface**

Manifold scripts can control user interface elements such as dialogs. A script that controls the user interface must run in a separate thread, which can be arranged via the Properties context menu that pops up when we right click on a script in the project pane and choose Properties. This menu allows us to check the Run in separate thread property. Running a script in a separate thread allows using advanced scripting techniques such as scripting the user interface, but might be slightly slower in some cases. By default, the option is turned off. See the User Interface Scripting topic.

**Notes**

Visual Basic .NET is often abbreviated "VB .NET" within Manifold dialogs, apparently inserting an extra space between the "VB" and the " .NET." This usage follows the most common Microsoft practise. However, Microsoft is so large that there is not a totally uniform usage within the company and some Microsoft publications will use "VB .NET", "VB.Net" and even "VB NET." This documentation tends to use VB .NET.

Within Manifold, all .NET assemblies (including Manifold.Interop) are installed into the global assembly cache to facilitate reuse.

**Internet Resources**

Visit the Manifold website at http://www.manifold.net and drill down into the Free Stuff page for links to scripting and programming examples.

Visit the Support page on the Manifold website to find links to the latest Manifold User Group forums in which scripting and programming are discussed. It is extremely important to tap into the collective experience of the Manifold user community when first learning to program Manifold System. Seeing examples and reading discussion on programming is a very important part of learning.

The easiest forum to access is the Georeference forum originally started by the Manifold System User Group and now run by the Manifold online community. Click on the Community page at the manifold.net home page or simply visit http://forum.manifold.net - Thousands of Manifold users from all over the world meet at that forum to discuss Manifold System, programming, IMS and related topics.

**See Also**

Developing Applications
Activation Keys and Serial Numbers

Scripts
Forms
Debugger
Programming Reference
Developing Applications

Manifold applications may be developed using a variety of approaches, from simple scripts within Manifold, to code that functions as add-ins within Manifold, to standalone applications utilizing Manifold through the object model, to web applications. See the Programming Manifold topic for an introduction to the range of possibilities.

This topic provides some supplemental notes for programmers creating external applications which utilize Manifold through the object model, via COM or .NET.

Licensing

Every machine than runs Manifold or any part of Manifold must have a Manifold license. A purely external application that utilizes Manifold only through the object model needs only a Manifold runtime license. At the present writing there are two Manifold runtime licenses available:

- **Professional Runtime x64** - This license provides access to most Manifold API features and is used for applications that do not need Enterprise Edition features or features in Manifold extension options.
- **Universal Runtime x64** - This license provides access to all Manifold API features including Enterprise Edition, Business Tools, Geocoding Tools and Surface Tools features. It is used in cases where Enterprise features such as access to a spatial DBMS like Oracle Spatial are required or when some specific feature within an extension is required.

Note that an x64 license will run in 32-bit Windows as well as 64-bit Windows. See the 32-bit and 64-bit Manifold Editions topic for a discussion of 32-bit and 64-bit licenses and issues.

Volume discounts are available for runtime licenses. See the manifold.net website for the volume discount price schedule. See the Advice for Applications Developers section in the Programming Manifold topic for a discussion of serial number and activation issues of interest to developers.

Learning and Support

Microsoft Visual Studio is the most common development tool used for developing applications that utilize the Manifold API. Visual Studio is used to create Manifold itself and is the only development environment supported by Manifold Technical Support when processing developer support incidents. It is therefore wise for programmers who wish to have maximum support resources from manifold.net and from other programmers in the Manifold user community to utilize Visual Studio as well. Visual Studio Express editions are available for free download from Microsoft.

General tips for programmers developing their first applications using the Manifold API:

- **Gain proficiency with Visual Studio** - Programming a sophisticated, complex and extensive API like Manifold is not a project for beginners. The better your programming skills are in general and the greater your expertise with Visual Studio, the smoother your projects with Manifold will go.
- **Master accessory technologies** - Many applications will involve other software: web applications, for example, will require the usual web development skills such as precise familiarity with IIS, administrative and security expertise and so forth. Almost all sophisticated web applications and many standalone applications will utilize a DBMS. Solid mastery of such accessory technologies will help avoid confusion while learning Manifold.
- **Learn Manifold as an interactive application** - Manifold is an immense application with numerous controls that help manage the intrinsic sophistication and complexity of GIS. If you don't learn Manifold interactively you won't have the conceptual and practical foundation to use the API effectively.
- **Study documentation** - Information of interest to programmers will often be found in the documentation within topics aimed at interactive users as well as within the programming reference.
- **Study examples** - Carefully study all examples published on the manifold.net web site, in the documentation, on other sites and in the forum.
- **Use the forum** - Become a master at using the GeoReference forum, at http://forum.manifold.net as of this writing. Follow threads, learn to search deep into the forum, study examples and contribute to discussions on Manifold programming in the forum.
- **Leverage tech support** - Read the Technical Support topic and the Support page on the manifold.net web site. Programmers who have mastered Visual Studio, who have learned Manifold interactively, who read documentation and study examples effectively and who participate in the forum
will usually never seek technical support. Nonetheless, should you need to use a developer technical support incident it is important to use that incident effectively.

The above assumes that programmers have a basic foundation in those GIS concepts they will need that cannot be learned through Manifold documentation. If a programming project involves sophisticated manipulation of projections, for example, it could be that the developer will need some supplemental education through distance learning, use of a consultant or attendance at a local university course.

**Updates and New Releases**

Manifold frequently issues routine updates to existing releases and every year or so issues a major new release. Programmers upgrading either to an update or to a new release should carefully study the release notes for the update or new release, as small changes (or major changes) may require rewriting of code for compatibility.

In general, upgrading between major versions of Manifold, for example, from Release 7x to Release 8, requires a recompilation but upgrading within routine updates, such as any updates published for Release 8, does not.

Following are step-by-step examples of using Visual Studio to upgrade applications in typical scenarios involving a new Manifold release:

**Upgrading a console application or class library written in C#, that only uses Manifold.Interop.dll:**

1. Launch Visual Studio, open the project, open the Solution Explorer pane, open the References node for the project,
2. Remove the reference for Manifold (right-click the reference and select Remove),
3. Right-click the References node, select Add Reference, switch to the COM tab, select Manifold System Type Library, click OK,
4. Rebuild.

**Upgrading a Windows application written in C#, that uses the Manifold ActiveX control:**

1. Remove both the bin and obj folders
2. Launch Visual Studio, open the project, open the Solution Explorer pane, open the References node for the project,
3. Remove the references for AxManifold and Manifold,
4. Right-click the project node, select Add - Windows Form, click Add, open the Toolbox pane,
5. Drop any old items for the Manifold ActiveX control you might have,
6. Add an item for the currently installed version of the Manifold ActiveX control (right-click the empty area in the toolbox, select Choose Items, switch to the COM Components tab, check Manifold MapControl Object, click OK),
7. Drop the item for the Manifold ActiveX control onto the form (to re-generate the dropped references),
8. Switch to the Solution Explorer pane, remove the new form,
9. Rebuild.

**Upgrading a console application or class library written in VB.NET, that only uses Manifold.Interop.dll:**

1. Launch Visual Studio, open the project, open the Solution Explorer pane, double click the My Project node,
2. Switch to the References tab, remove the reference for Manifold,
3. Click Add and select Reference, switch to the COM tab, select Manifold System Type Library, click OK,
4. Rebuild.

**Upgrading a Windows application written in VB.NET, that uses the Manifold ActiveX control:**

1. Remove both the bin and obj folders,
2. Launch Visual Studio, open the project, open the Solution Explorer pane, double click the My Project node, switch to the References tab,
3. Remove the references for AxManifold and Manifold (which will turn blank after you remove AxManifold - that's OK).

4. Right-click the project node, select Add - Windows Form, click Add, open the Toolbox pane, drop any old items for the Manifold ActiveX control you might have,

5. Add an item for the currently installed version of the Manifold ActiveX control,

6. Drop the item for the Manifold ActiveX control onto the form,

7. Switch to the Solution Explorer pane, remove the new form,

8. Rebuild.

Many of the mechanical details of creating and maintaining applications are specific to the choice of a particular development environment such as Visual Studio, as can be seen in the examples above. Programmers are therefore strongly encouraged to use Visual Studio to get the benefit of as many support and informational resources as possible.

See Also

Programming Manifold

Scripts

Scripts are Manifold programs written in an ActiveX or .NET programming language that is installed on the computer system.

Visual Basic Scripting Edition is the default ActiveX scripting language used with Manifold; however, JScript (Microsoft's implementation of Javascript) is also frequently used. Scripts may be written in any language for which an ActiveX scripting engine is available. Some users, for example, prefer to write scripts in Python or Perl using ActiveX scripting engines for those languages that are available from third party suppliers.

Documentation and technical support for scripting in Manifold uses VBScript with occasional examples in JScript. The use of other languages is respected but is not covered by documentation or technical support. VBScript and JScript are covered since these are standard Microsoft languages that may be redistributed by manifold.net with Manifold.

Manifold also supports scripting in Microsoft .NET languages such as C#, JScript .NET, IronPython .NET and VB .NET.

Scripts are components that appear in the project pane.

- To create a script, choose File - Create - Script or using the Create button in the project pane toolbar, choose Script.
- Double click a script component to edit it in a script window. Specify the language to be used with Scripts - Language.
- Highlight a script in the project pane and press the Run button to execute it.

When a script window is open, the Script menu is available:

- Open Form
  Open the form associated with this script.

- Open Table
  Open the table associated with this script.

- Run
  Run the script

- Run under Debugger
  Run a script under control of the debugger. Available if the Debugger is installed.

- Pause
  Pause execution of a script. Available if the Debugger is installed.

- Stop
  Stop execution of a script. Available if the Debugger is installed.
Step Into  Step over a routine. Available if the Debugger is installed.

Step Out  Step into a routine. Available if the Debugger is installed.

Step Over  Step out of a routine. Available if the Debugger is installed.

Language  Specify a scripting language. Available choices will include all supported ActiveX scripting engines installed on the computer system as well as all supported .NET scripting languages.

References  View and modify script references. Available for .NET scripts.

Compile to DLL  Compile a script into a DLL. Available for .NET scripts.

Note: In the above comments, "Available if the Debugger is installed" means the capability is available if an ActiveX language is being used and the necessary Microsoft facilities to enable the Manifold debugger have been installed. See the Debugger topic.

Use keyboard shortcuts to set and clear breakpoints. Click on a text line in the script window and then press F9 to toggle a breakpoint at that line.

Example

Choose File - Create - Script to create a new script component in the project pane.

The Create Script dialog box will allow us to name our new script, provide an optional description and choose a language (default choices being C#, JScript, JScript.NET, VB.NET and VBScript). If a .NET language is selected, we can check the Add references for standard .NET modules box. Regardless of whether or not you check this box, you can always add new references or remove existing references using the Script - References command.

The new script component will appear in the project pane marked with a script icon. Clicking on the component to highlight it will provide information about script in the project pane status bar, including the size of the script and the language used.

Double-click the script component to open it in a script window. The new script will appear with startup text appropriate for that language, by default showing a simple script to print "Hello, World!" to a message box.

Running a Script
Creating a new script automatically loads it with code for the language of choice to display the classic "Hello, World!" message in a message box. This helps us get started by providing a framework and is extremely useful when writing applications that say "Hello."

Suppose we have created a VB .NET script. The default text in the script will be:

```
Imports Manifold.Interop.Scripts

Class Script
    Shared Sub Main
        Context.Application.MessageBox("Hello, World!", "Script")
    End Sub
End Class
```

Note that the above script uses the built-in Application object.

Press the Run button in the Tools toolbar to run the script. We can also click onto the script component in the project pane to highlight it and then press the Run button in the project pane.

A message box appears with our desired message. Press OK to close the message box.

If we had used a .NET language like VB .NET, whenever we run the script it will first be compiled, which may add a delay before the script launches in the case of large scripts. We can avoid this delay by using the Script - Compile to DLL command to compile the script into a DLL.

Any errors detected during compilation of a .NET script will be displayed in the Errors pane.

Note: If we are working in an ActiveX language and the Debugger has been configured we can run scripts using the Debugger. See the Debugger topic for more information.

**Command Line Script Startup**

The `autoexec:<scriptname>` command line switch allows running a script after opening the MAP file specified in the same command line. For example, the following command line will open the `mymap.map` project file and then run the `myscript` script found in that project file.

```
manifold.exe /autoexec:myscript c:\MyProjects\mymap.map
```

**Saying Hello in C#**

The C# version of "Hello, World" is:
using Manifold.Interop.Scripts;

class Script {
    static void Main()
    {
        Context.Application.MessageBox("Hello, World!", "Script");
    }
}

Saying Hello in JScript

The JScript version of "Hello, World" is:

function Main()
    {
        Application.MessageBox("Hello, World!", "Script");
    }

Saying Hello in JScript .NET

The JScript .NET version of "Hello, World" is:

import Manifold.Interop.Scripts.*;

class Script {
    static void Main()
    {
        Context.get_Application().MessageBox("Hello, World!", "Script");
    }
}

Saying Hello in VB .NET

The VB .NET version of "Hello, World" is:

Imports Manifold.Interop.Scripts

Class Script
    Shared Sub Main
        Context.Application.MessageBox("Hello, World!", "Script")
    End Sub
End Class

Saying Hello in VBScript

The VBScript version of "Hello, World" is:

Sub Main
    Application.MessageBox "Hello, World!", "Script"
End Sub

Saying Hello in IronPython
IronPython is the fine .NET language created by Jim Hugunin and supported by Microsoft. It has emerged as the Python of choice for Microsoft .NET and is supported by Manifold Technical Support for developer level support incident questions.

```python

def Main():
    Application.MessageBox("Hello, World!", "Script")

Main();
```

The above example consists of four lines of code, the third line of which is blank. Whitespace is significant in Python so it is important that the third line remain blank for the example to function.

IronPython scripts intended to be used for active columns should only include the body for a single script function.

Creating a script in IronPython automatically adds references to the IronPython assemblies. To work correctly in all cases, the IronPython assemblies should be installed in the Global Assembly Cache. The Global Assembly Cache (GAC) is a central place for .NET assemblies on a machine. Putting an assembly into the GAC makes it possible for all .NET applications to locate and use that assembly.

To put IronPython assemblies into the GAC, open the Command Prompt as a user with administrative privileges (use Run As Administrator in Windows Vista), navigate to the IronPython folder, and run the following commands:

```cmd
    gacutil -i ironpython.dll
    gacutil -i ironmath.dll
```

The above requires installation of the .NET SDK (which contains the GACUTIL program). The .NET SDK is easily found and downloaded from the Microsoft web site.

Thanks go out to Jim Hugunin for creating IronPython (excellent name!) and also to Microsoft for supporting IronPython. Well done!

**Saying Hello in PythonScript and PerlScript**

Many languages are available as ActiveX scripting languages for Windows. For example, installing the Active State Python or Perl distributions (free as of this writing from http://www.activestate.com) will add Python and Perl scripting capabilities to Manifold. As with JScript, scripts in these languages use the built-in Application object.

**PythonScript:**

```python
def Main():
    Application.MessageBox("Hello, World!"

Main();
```

**PerlScript:**

```perl
sub Main {
    $Application->MessageBox("Hello, World!");
}
```

Some users prefer scripting in Python or Perl to scripting in VBScript or JScript; however, Manifold technical support is able to support only VBScript, JScript and the .NET languages.

**Adding Scripts as Toolbar Buttons or Menu Commands**

Scripts (which may call external COM objects) may be added as custom toolbar buttons or menu items. See the Add-Ins topic for information on adding new toolbar buttons or menu commands.

**Script Window Keyboard Shortcuts**
Script windows support well-known Windows keyboard shortcuts, including those often used in Windows programming editors:

- **CTRL-A** Select All
- **CTRL-X** Cut selected text and copy to the Clipboard.
- **CTRL-C** Copy selected text to the Clipboard.
- **CTRL-V** Paste text contents of the Clipboard.
- **CTRL-L** Delete current line.
- **CTRL-K** Comment out selected text.
- **SHIFT-CTRL-K** Uncomment selected text.
- **CTRL-M** Convert selected text to lower case.
- **SHIFT-CTRL-M** Convert selected text to upper case.
- **CTRL-Insert** Equivalent to **CTRL-C**. An old Windows shortcut kept for compatibility.
- **SHIFT-Insert** Equivalent to **CTRL-V**. An old Windows shortcut kept for compatibility.
- **CTRL-Z** Multi-level Undo.
- **CTRL-Y** Multi-level Redo.
- **CTRL-[]** Check the character near the cursor and, if it is a bracket character, jump to the matching bracket.
- **CTRL-SHIFT-[]** Check the character near the cursor and, if it is a bracket character select the text between the bracket and the matching bracket.
- **SHIFT-Delete** Same as **CTRL-X** or Cut.
- **ALT-Backspace** Same as **CTRL-Z** or multi-level Undo.

**CTRL-K** and **SHIFT-CTRL-K** comment/uncomment commands will automatically switch between commenting styles based on the language specified for the script window with the Script - Language command. Works for C#, JScript, JScript .NET, PERL, Python, VBScript and VB .NET.

Recognized bracket combinations for use with **Ctrl-[]** and **Ctrl-SHIFT-[]** are { and }, [ and ], ' and '”, and # and # (used to delimit dates in queries).

**Example: Referencing a Script that has been Compiled into a DLL**

The Script - Compile to DLL command may be used with .NET languages to avoid the slight compile delay when the .NET script is compiled before running. The command may also be used to compile .NET language scripts into DLLs for future use in other scripts, as in the following example:

1. Launch Manifold. Create a new project.

2. Create a new script using VB .NET. Enter the following script text:

```vbnet
Imports Manifold.Interop
Public Class Test
    Public Shared Function ComponentCount(D As Document) As String
```

Public Class Test
    Public Shared Function ComponentCount(D As Document) As String
Return D.ComponentSet.Count
End Function
End Class

3. Choose Script - Compile to DLL and save the DLL into the Manifold installation folder (that is, the Windows folder into which Manifold System itself was installed, by default, C:\Program Files\Manifold System).

4. Create another script using VB .NET. Invoke Script - References. Click New, and select the compiled DLL created in the above steps. Enter the following script text:

Imports Manifold.Interop.Scripts

Class Script
    Shared Sub Main
        Context.Application.MessageBox( _
        Test.ComponentCount( _
        Context.Application.ActiveDocument), "Script")
    End Sub
End Class

Invoke Script - Run. The script will show the number of components in the currently opened MAP file.

It is, of course, possible to place a compiled DLL into a folder that is different from the Manifold installation folder, but then we would have to make sure the .NET loader will be able to find both:

a) the compiled DLL when running a Manifold script that refers to it (usually achieved by either signing the compiled DLL and putting it into the global assembly cache, or by putting the compiled DLL into a folder mentioned in the PATH environment variable), and

b) all Manifold DLLs required when loading the compiled DLL (usually achieved by using a custom configuration file).

Running a .NET script first searches for referenced assemblies that use relative paths in the Manifold installation folder and then searches in the Configuration folder set in Tools - Options in File Locations.

See Microsoft documentation on .NET languages for details, including MSDN. In particular see this MSDN topic (URL current as of the time of writing) for more information:


Notes

Attempting to save or close a project in the middle of an executing script will display a warning message. Stop any executing scripts before saving or closing a project.

Script windows support multi-level Undo / Redo. One can CTRL-Z (Undo) backwards through many changes and CTRL-Y (Redo) forwards to redo many Undo operations.

Script windows will show their contents using different colored font for different parts of the script. This is called syntax highlighting and is a helpful way of identifying typographical errors. Fonts used in script windows may be specified in Tools - Options. However, fonts used with script windows will be restricted to fixed-width fonts.

See Also

Add-Ins
Script Examples
See the Active Columns topic for a cool use of scripts within tables. See Active Columns using VBScript and Active Columns using JScript for examples using active columns and script functions written in VBScript and JScript.

See the Forms and Scripts topic for examples of using scripts together with forms.

See the Debugger topic if you are working with ActiveX languages and interested in using the Debugger.
Click Events
Manifold scripts may be written for processing of mouse clicks and double clicks done in drawing, image, labels, map, profile, surface, or theme windows. Click events are routed to a script or form named OnClick. Double click events are routed to a script or form named OnDoubleClick. The parameters of both events are delivered through the global EventArgs object.

The Process global events with scripts option in Tools - Options - Miscellaneous allows (the default) or disallows the processing of click and double click events with scripts.

If desired, the OnDoubleClick event handler can cancel further processing of a clicked object (which might otherwise go to an URL or result in the display of the Object Fields dialog) by setting the Handled property of the EventArgs object to True.

Example
Create a VBScript script named OnClick and enter the following script:

```vbscript
Sub Main
    If EventArgs.HasObject Then
        Application.MessageBox EventArgs.Object.ID, "Clicked Object"
    End If
End Sub
```
Create a new drawing and populate it with some objects. Click an object. The system will display a message box reporting the object's ID.

See Also
EventArgs Object
Tools - Options
Update Batching

The BatchUpdate property of a Document object turns batch update mode on and off. The Document object manages components and document-wide properties. Batch update mode controls how changes made by a script are propagated to the mapfile.

When BatchUpdate is off (the default), any changes made in the script are immediately propagated to the mapfile. This is convenient for many purposes and assures that script objects always report fresh information. However, at times making a update on every change will hurt performance. For example, when selecting pixels in a surface or image one loops through program logic that sets the pixel mask for each pixel. Propagating each change would mean a refresh of the entire display for each pixel. It would be best in such cases to turn BatchUpdate on, run the selection logic, and then turn BatchUpdate off.

When batch mode is on, changes made by a script are not immediately propagated to the mapfile. This is faster but can make script objects report outdated information. The recommended technique is to turn BatchUpdates on when bulk changes are being made to data and to then turn it back off when all necessary changes have been done. This is especially important with operations on images and surfaces, but also makes sense for selection operations with objects in drawings.

Example

Suppose we are working with pixels in a surface. We will select or deselect each pixel based on some criterion. The following VBScript script fragment shows use of BatchUpdate with some lines boldfaced for emphasis:

```
Sub Main
    Set app = Application

    ' -- locate opened window and ensure it displays surface
    Set window = app.WindowSet.ActiveWindow
    Set component = window.Component
    If window.Component.Type <> ComponentSurface Then
        MsgBox "No surface."
        Exit Sub
    End If

    Set surface = window.Component
    Set pixels = surface.PixelSet

    ' -- miscellaneous program logic goes here

    ' -- turn batch updates on
    app.ActiveDocument.BatchUpdates = True

    ' -- select pixels
    For i = 0 To pixels.Count-1
        ' -- pixel selection logic goes here, for example,
        ' -- pixel.Mask = pixel.Mask Or 1 to select and
        ' -- pixel.Mask = pixel.Mask And Not 1 to deselect.
        Next

    ' -- turn batch updates off
    app.ActiveDocument.BatchUpdates = False

End Sub
```

Note that all program logic except the bulk selection process done in a loop is done with BatchUpdates turned off. Only the pixel-by-pixel selection loop has BatchUpdates turned on.
See Also

Scripting Reference
Document Object
User Interface Scripting

Manifold supports scripting of user interface elements such as dialogs and controls. User interface scripting is available for regular scripts and forms, but is unavailable for active columns, where it would be impractical for performance reasons, external applications and web sites using Manifold objects (where there is no Manifold user interface present).

Scripting of the user interface allows running interactive commands and controlling dialogs and controls, with scripting supported for common Windows controls such as buttons, check boxes, labels, text boxes, list boxes, combo boxes, tree views and list views. User interface scripting objects support scripting of the progress bar and slider controls, via the Text property. All Manifold dialogs support user interface scripting.

Scripts that operate with user interface elements must be run in a thread separate from the user interface thread. This can be arranged via the Properties context menu that pops up when we right click on a script in the project pane and choose Properties. This menu allows us to check the Run in separate thread property. Running a script in a separate thread allows using advanced scripting techniques such as scripting the user interface, but might be slightly slower in some cases. By default, the option is turned off so this property must be explicitly set to enable user interface scripting.

Form scripts can be configured to run in a separate thread, which allows scripting the user interface. Running a script from another script spawns a new thread, in case the callee is configured to run in a separate thread. This allows freely mixing scripts that are configured to run in the calling thread or a new thread.

User interface scripting objects support numerous Manifold facilities, including:

- Scripting panes in the main window as well as scripting toolbar controls. Each toolbar button is interpreted as a separate user interface control.
- Scripting combo boxes, text boxes and other non-button controls in toolbars.
- Support for "pushing" buttons in the drop-down portions of color, size, and style wells, such as buttons for invoking the thematic formatting dialog, and similar windows.
- Support for scripting format lists such as the list of colors in the Display Options dialog for surfaces.
- Invoking a user interface command within a script allows specifying the command context to access commands available in the various context menus and in panes.
- User interface scripting objects support reading and writing values of items in list view controls, such as the list view in the contouring dialog.

User interface scripting objects also support scripting non-Manifold dialogs, such as the open file dialog. Names of controls for non-Manifold dialogs are synthesized on the fly.

Example - Spatial Overlays

Suppose we have a map component called Map that includes a drawing of areas named States and a drawing of points named Cities. The drawings share some common columns. We can script a spatial overlay with the following:

Sub Main
    Document.ComponentSet("Map").Open
    Set ui = Application.UserInterface
    ui.InvokeCommand "MapSpatialOverlay"
    Set dlg = ui.ModalDialog
    dlg.ControlSet("ComboBoxSource").Text = "[All Objects in States]"
    dlg.ControlSet("ComboBoxTarget").Text = "[All Objects in Cities]"
    dlg.ControlSet("ComboBoxMethod").Text = "Areas to contained points"
    dlg.Accept
End Sub

Discovering Control Names
It’s easy to discover the names assigned to various controls in Manifold dialogs. Right clicking the caption of a dialog or undocked pane or toolbar and choosing List Controls from the context menu will display a list of programmatic names of dialog or pane controls or toolbar commands that may be used with user interface scripting.

Another choice is to use the /clist command line option, which dumps the names of all commands available for user interface scripting using the InvokeCommand method of the UserInterface object into a text file.

Example - Discover Control Names in the Dissolve Dialog

If we open a drawing and then choose Drawing - Dissolve we open the Dissolve dialog.

Right clicking on the caption for the dialog...

...and then choosing List Controls from the context menu...
Programming

...will display the Controls list of the names of controls available for programmatic use.
Tech Tip
Attempting to locate a pane using its name within a script allows omitting the component name postfix. For
example, attempting to locate a pane named Control Points will succeed even if there is an active component,
which sets the pane name to Control Points - [Component Name].
See Also
Scripts

Script Examples

Script Examples
See the Manifold web site for scripting examples that may be published from time to time in addition to those
below.
.NET Language Examples
Manifold may be programmed in .NET languages such as VB .NET or C#. Some elementary examples follow.
Example
Display all drawings found in the active project using VB .NET:
Imports Manifold.Interop
Imports Manifold.Interop.Scripts
Imports Microsoft.VisualBasic
Class Script
Shared Sub Main
Dim doc As Document = Context.Application.ActiveDocument
Dim cmp As Component
Dim rpt As String = ""

2393


' traverse all components and add name of each drawing to
' report
For Each cmp In doc.ComponentSet
    If cmp.Type = ComponentType.ComponentDrawing Then
        If rpt <> "" Then
            rpt = rpt & vbCrLf
        End If
        rpt = rpt & cmp.Name
    End If
Next

' check if there are no drawings
If rpt = "" Then
    rpt = "No drawings."
End If

Context.Application.MessageBox(rpt, "Script")
End Sub
End Class

Example

Do the same using C#:

using System;
using Manifold.Interop;
using Manifold.Interop.Scripts;
class Script {
    static void Main() {
        String rpt = "";

        // traverse all components and add name of each drawing to
        // report
        foreach (Component cmp in doc.ComponentSet) {
            if (cmp.Type == ComponentType.ComponentDrawing) {
                if (rpt != "")
                    rpt = rpt + "\n";
                rpt = rpt + cmp.Name;
            }
        }

        // check if there are no drawings
        if (rpt == "")
            rpt = "No drawings."

        Context.Application.MessageBox(rpt, "Script");
    }
}

Example
Create a new drawing and populate it with a point object using VB .NET:

Imports Manifold.Interop
Imports Manifold.Interop.Scripts

Class Script
    Shared Sub Main
        Dim app As Application = Context.Application
        Dim doc As Document = app.ActiveDocument

        ' create new drawing
        Dim drw As Drawing = doc.NewDrawing("New Drawing")

        ' create new point and then new object
        Dim pnt As Point = app.NewPoint(10, 20)
        Dim pntGeom As Geom = app.NewGeom(GeomType.GeomPoint, pnt)
        drw.ObjectSet.Add(pntGeom)

        ' open created drawing
        drw.Open
    End Sub
End Class

See Also

See the manifold.net web site for additional scripting examples.

Forms

Forms are Windows dialogs created within Manifold that provide customized user interfaces to scripts, queries, tables and other items. Forms are themselves programming object that have properties and methods and can react to events. When a form is created, a script for that form is automatically created as well. The scripting code written in the script specifies the program logic that is used with the controls on the form.

For example, a script that showed the positions of trucks might utilize a form that included the controls shown above to allow inexperienced users to choose which class of trucks they wished to see on the map. Manifold forms can include a very wide array of controls, including toolbars, status bars, hierarchical tree diagrams, iconic data browsers, rich text panes, multimedia controls and much, much more.

Forms play the same role in Manifold scripting using Visual Basic scripting that they do in development of Visual Basic Applications within Visual Studio. In fact, almost all of the MSDN and Visual Basic documentation from Visual Studio may be used with Manifold System. Manifold's form design and form editing capabilities are similar to those of Visual Studio.

For a quick introduction to Windows programming, event-driven programming and forms-based programming, see the Programming Manifold introductory topic.
Creating a Form

1. Use the File - Create - Form command to add a form to a project. Use a template if desired.
2. Click open the form.
3. Resize the form as desired and set its Form Properties, such as the caption for its title bar.
4. Add controls from the Tools toolbar and the Tools (Advanced) toolbar.
5. Arrange controls by selecting them and using the Alignment toolbar.
6. For each control, specify its Properties.
7. Open the script associated with the form and write code to implement program logic and control behavior.

Forms may be edited using the methods described in Editing Forms. For example, controls from other forms may be dragged and dropped into a form.

Forms Context Menu

Right click onto any unused part of a form to call up the context menu for forms. Right clicking onto a control in a form will call up the context menu for that control.

- **Size to Fit**  
  Resize the form so it just encloses the controls that are in it.

- **Test**  
  Execute the form for placement of controls. This is not a full test of the script and only provides a visual preview. Use Alt-F4 to close the form if it is not equipped with a Control Box "close" button in the upper right corner.

- **Properties**  
  See and set the form's properties.

Tech Tips

Forms may be copied and pasted. Forms may also be imported from other .map files using the File - Import - Component command. This allows us to save collections of frequently used forms in a .map file from which a desired form may be imported when needed.

Although forms cannot be shared on an Enterprise server if Enterprise Edition is being used, there is an alternative way of creating reusable forms using .NET scripts: The Script - References dialog is available whenever a script window using a .NET language as the Script - Language is opened. The dialog allows specification of external .NET modules to which the script may refer. For example, one can refer to System.Windows.Forms.dll and have a script display a form. The resulting script can then be shared on an Enterprise server and reused in other projects.

See Also

- Form Properties
- Form Controls
- Editing Forms
- Scripts
- Script Examples
- Script - References
- Programming Manifold
Form Properties

Forms are Windows dialogs created within Manifold that provide customized user interfaces to scripts, queries, tables and other items. Forms are themselves programming objects that have properties and methods and can react to events. See the Programming Manifold topic for an introduction to programming and the Forms topic for a brief introduction to forms.

Form Properties

Right click onto a form (except directly onto a control) and choose Properties from the context menu to see and set the form’s properties:

- **Caption**: The text string for the title bar.
- **Tag**: A string available for free use by the programmer. Often used for comments.
- **Control Box**: Check the Control Box to create the X “close” box in the upper right corner of the form and to enable creation of Minimize and Maximize boxes. If checked, will add to the title bar a standard Windows right mouse context menu with Restore, Move, Size, Minimize, Maximize and Close (ALT+F4) choices.
  
  **Note**: if you create a form with no Control Box there is no obvious way to close it with the mouse when it is being tested. Use ALT F4 to close it.
- **Maximize Box**: Create a maximize box control in the title bar along with the control box.
- **Minimize Box**: Create a minimize box control in the title bar along with the control box.
- **Center on Screen**: Display the form in the center of the main Manifold window.
  
  **Left**: Left margin position of the form, if Center on Screen is not checked.
  
  **Top**: Top margin position of the form, if Center on Screen is not checked.
- **Width**: Width of the form.
- **Height**: Height of the form.

**Very Important**: If you uncheck the Control Box and then use Form - Test there will be no way to close the form except by doing an ALT-F4. Even if controls (such as a "Close" command button) are added to the form to close it without having to use the X control box in the upper right corner, those controls will not be functional in "test" mode. They will only work if the form is run using the Run button from the project pane.

The form property sheet also has standard Color and Font property pages used to customize colors and font used in the form.

Form Properties Set Programmatically

Forms are created with the most frequently used defaults for properties. In addition to the above properties that are changed through the Properties dialog, several more properties may be set programmatically within the form’s script.

- **Border**: FixedSingle (default) or None. None results in an uncaptioned rectangle of background color and will also cause minimize, maximize and system menu settings to be turned off.
  
  **Note**: if you create a form with no border there is no obvious way to close it with the mouse when it is being
tested. Use **ALT F4** to close it.

**Pointer**  
The type of mouse pointer displayed when over the form. Choose from 16 different types plus a custom pointer

**Visible**  
Make this form visible or invisible.

**Enabled**  
True or False. True means the form can respond to user-generated events, false prevents it from responding. We can disable a form if we wish it to display information in a read-only way.

### Form Properties for Controls

Right click onto a control in a form and choose **Properties** from the context menu to see and set properties for the control.

Manifold adds an extra tab, the Control tab, to the usual tabs that appear in the design time properties dialog for controls. This tab provides access to the properties for this control maintained by the form. This extra tab will appear at design time for all controls. Although strictly speaking these are properties of the form that is the container for the control and not of the control, they are easiest to specify at design time if they appear within each control’s properties dialog.

**Name**  
The name used by the form to identify this control.

**Tag**  
A string available for free use by the programmer. Often used for comments.

**Visible**  
Make this control visible or invisible.

**Left**  
Left margin position of the control.

**Top**  
Top margin position of the control.

**Width**  
Width of the control.

**Height**  
Height of the control.
Form Controls

The Tools toolbar and Tools (Advanced) toolbars are used with Forms to create Windows standard user interface controls. These toolbars are enabled when a Form is the active window. Forms are used to provide user interfaces to scripts, to view tables and for other functions within Manifold.

**Very Important:** Common ActiveX controls for forms are available only for use with scripting in Manifold System running in 32-bit mode. Common ActiveX controls for forms are *not* available when Manifold is running in 64-bit mode in 64-bit Windows systems. See the 32-bit and 64-bit Manifold Editions topic when running 64-bit Manifold System editions.

To add a control to a form, click open the form and then click on the control desired. Click and drag within the form to create the control in that location at the given size. Then, right-click on the control and set its properties as desired. When controls have been created in a form they may be rearranged as desired, aligned neatly and otherwise edited as described in Editing Forms. Manifold’s built-in forms editing uses Visual Studio style mouse moves coupled with the Selection toolbar and other cool Manifold methods.

As in Visual Studio language form creation, the tab order of controls is set by default by the order in which they were created. This may be adjusted at will by right clicking and choosing move up, etc.

**Sources of Controls**

Manifold can use any ActiveX control within scripts. There are several classes of controls that have been provided by manifold.net or that may be used within Manifold.

- Frequently used controls are hosted on the Tools toolbar.
- Advanced controls are hosted on the Tools (Advanced) toolbar.
- Numerous accessory controls are provided in various .ocx collections installed by Manifold for your use.
- Manifold scripts can use controls found in any .ocx installed on your machine by other applications, for example, by Microsoft Office 2000.
- Numerous freeware and shareware ActiveX controls are available for free download via Internet.
- Many vendors sell collections of ActiveX controls for special purposes.

Manifold installs the full set of standard, redistributable controls shipped by Microsoft with Visual Studio Enterprise Edition, plus all standard, redistributable controls made available by Microsoft to its ISV partners via the Microsoft Developer Network. Users of Manifold’s Visual Basic Scripting system therefore have the same access to redistributable controls as do developers working at the highest level with these professional-quality Microsoft tools.

Manifold includes everything you need to create new programs within Manifold in Visual Basic scripting or Javascript using the above controls. There is no need to purchase Visual Studio or any extra language. Of course, if you do have Visual Studio and tools such as Visual C++ and Visual Basic you can use those as well. We highly recommend Visual Studio for professional development: it is the environment within which Manifold itself was created using Visual C++.

**Documentation for Controls**

One major advantage of using Microsoft technology is the ease with which standard controls may be included within an application such as Manifold. A major benefit to the use of standard controls is the wide availability of comprehensive reference texts devoted to Visual Basic programming using those controls. The manifold.net team has included many very sophisticated controls within Manifold for your use. These are not documented in detail, since doing justice to the power and flexibility of these controls requires a dedicated text. Programmers using these controls should have at hand one of the many excellent texts on programming in Visual Basic that covers the various controls in detail. This documentation only covers those properties (in brief) that may be set at design time via the control's Properties dialog.

To give an idea of the comprehensive power of the controls included in Manifold, we have included a summary listing of run time properties in the RichTextBox control topic together with a summary listing of this control's methods. The RichTextBox control may be used to create a full-featured, professional quality word processor within a Manifold script, if so desired. Because it is a data bound control, a more frequent use in a GIS environment might be to create elaborate forms, reports, or other richly-formatted documents that automatically load text data from a variety of external data sources.
Properties pages for some controls will feature an enabled Help button. The Help button is linked to Microsoft Help for these Microsoft controls. If you have a Microsoft programming environment and associated Help documentation (like MSDN) installed, the Help button will launch the appropriate Microsoft Help topic. If you do not have a Microsoft programming environment with Help for these standard Microsoft controls, the Help button will not be able to launch any help pages.

Frequently Used Controls

<table>
<thead>
<tr>
<th>Tool</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Insert Check Box" /></td>
<td>Insert Check Box - Inserts check box controls.</td>
</tr>
<tr>
<td><img src="image" alt="Insert Option Button" /></td>
<td>Insert Option Button - Inserts option button controls. Only one option button in a group may be selected.</td>
</tr>
<tr>
<td><img src="image" alt="Insert Command Button" /></td>
<td>Insert Command Button - Inserts command button controls.</td>
</tr>
<tr>
<td><img src="image" alt="Insert Frame" /></td>
<td>Insert Frame - Draws frame with caption. Frames are used to set off groups of controls, often option buttons.</td>
</tr>
<tr>
<td><img src="image" alt="Insert Text Box" /></td>
<td>Insert Text Box - Inserts text box. Also called an edit field or edit control. This control can display text entered by the programmer at design time or entered by the user or assigned to the control by other code at run time.</td>
</tr>
<tr>
<td><img src="image" alt="Insert Static Text Box" /></td>
<td>Insert Static Text Box - Add a customized text box.</td>
</tr>
<tr>
<td><img src="image" alt="Insert List Box" /></td>
<td>Insert List Box - Inserts list box controls.</td>
</tr>
<tr>
<td><img src="image" alt="Insert Combo Box" /></td>
<td>Insert Combo Box - Inserts combo box controls.</td>
</tr>
<tr>
<td><img src="image" alt="Insert Horizontal Scroll Bar" /></td>
<td>Insert Horizontal Scroll Bar - Inserts horizontal scroll bar controls.</td>
</tr>
<tr>
<td><img src="image" alt="Insert Vertical Scroll Bar" /></td>
<td>Insert Vertical Scroll Bar - Inserts vertical scroll bar controls.</td>
</tr>
</tbody>
</table>

Advanced Controls

<table>
<thead>
<tr>
<th>Tool</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Insert Animation" /></td>
<td>Insert Animation - Plays silent .avi files to create dialog effects like the sheet of paper that flies between folders in the Windows copy progress dialog. Can also play silent .avi files dropped onto the control. Manifold provides 19 cool Windows dialog .avi files to get started.</td>
</tr>
<tr>
<td><img src="image" alt="Insert Chart" /></td>
<td>Insert Chart - Inserts a chart control.</td>
</tr>
<tr>
<td><img src="image" alt="Insert Image List" /></td>
<td>Insert Image List - Contains a collection of images that can be used by other Windows Common Controls such as ListView, TreeView, TabStrip and Toolbar controls, as well as with other controls with a Picture property. Saves development time by maintaining all images in a</td>
</tr>
</tbody>
</table>
standard, consistent catalog of images.

Insert Image Combo Box - Similar to standard combo but with ability to include pictures with each item in the list portion of the combo, with special facilities for list management.

Insert Date / Time Picker - Displays date and/or time info and provides an interface for modifying date and time. Dropdown menu provides a MonthView calendar.

Insert Month View - An easy way to view and set date information using a monthly calendar. Can select one or multiple dates or show up to 12 months at a time.

Insert List View - Displays data as ListItem objects with an optional icon. Four different views provide data as icons, small icons, a list or a report. A sophisticated control.

Insert Progress Bar - Create a progress bar display that can be used to show the progress of a process.

Insert Masked Text Box - Used to prompt users for input using a fixed format specified by a mask pattern, for example, a telephone number in a specified format. If a mask pattern is not used, this control behaves about the same as a standard text box.

Insert Rich Text Box - Allows entry and editing of text using advanced formatting features, such as paragraph formatting with left and right indents, hanging indents, bold and italic font and so on. Also supports object embedding using the OLEObjects collection. Write your own word processor or create elaborate forms for users.

Insert Slider - Insert a slider bar control that can be moved left and right.

Insert Tree View - Insert a hierarchical tree display. Designed to display data such as organization trees, entries in an index, files and directories on disk, etc.

Insert UpDown - Insert an increment/decrement control, also called spin buttons.

Insert Tab Strip - Insert a tabbed control where clicking on each tab brings it to the fore.

Insert Picture Clip - Inserts a picture clip.

Insert Multimedia Control - Manages Media Control Interface (MCI) devices like sound boards, MIDI sequencers, CD-ROM drives, audio players, videodisc players, etc. Add voice note recording to your dialog, or play spoken announcements.

Insert Status Bar - Insert a status bar like those at the bottom of many program windows. Complete with a nearly infinite set of properties for the intrepid programmer.

Insert Tool Bar - Insert a toolbar at the top of the form that hosts buttons or other controls. Yet another control with a vast array of properties.

Insert Cool Bar - Requires installation of Internet Explorer 3.0 or greater. Provides a modern, “railbar” look and the ability to create (get this!) user configurable toolbars like
those in IE.

- **Insert Common Dialog Control** - Provides a standard set of Windows dialog boxes for opening and saving files and selecting colors and fonts. Can also display Help.

- **Insert System Info Control** - Detects system events such as desktop resizing, resolution changes, time changes. Also provides operating system platform and version information and changes in AC/battery power status and Plug/Play hardware configuration. Used when writing applications for portable devices (GIS in the field) and for developing info for your tech support team.

- **Insert ActiveX Control** - Insert any ActiveX control available on this system.

**Other Controls**

Other controls from the various .ocx files registered on the system are available for your use. These include such cool controls such as Internet controls, winsock, comm controls and other fun objects that will dazzle your users and make your competitors regret they are not using Manifold.

Use the **Insert ActiveX Control** button in the **Tools (Advanced) toolbar** to draw a rectangle in the Form where the control should be placed. A dialog will pop-up showing all controls that are registered for use on this system from which the control may be selected.
Editing Forms

Forms are edited using a combination of commands from the Forms window itself, the Alignment toolbar, the Selection toolbar and by adding controls from the Tools and Tools (Advanced) toolbars. The appearance of the form and the controls it contains are specified by setting the Form Properties of the form and the properties of the controls it contains.

![Form Window](image)

Set the size of the form at any time by resizing it within the form window.

We can change the properties of the form by right clicking on the form and choosing Properties from the context menu.

![Form Properties](image)

Change the Caption property to change the caption in the title bar.

- Control Box
- Maximize Box
- Minimize Box

Check the Control Box to create the X "close" box in the upper right corner of the form and to enable creation of Minimize and Maximize buttons. Checking the Control Box option also equips the form's title bar with a right-click context menu consisting of the standard Windows choices of Restore, Move, Size, Minimize, Maximize and Close. Click the Minimize Box and Maximize Box boxes to add these controls to the title bar.

![Form Window](image)

The result will be shown in the form window. Note the changes in the title bar.
**Very Important:** If you uncheck the Control Box and then use Form - Test there will be no way to close the form except by doing an ALT-F4. Even if controls (such as a "Close" command button) are added to the form to close it without having to use the X control box in the upper right corner, those controls will not be functional in "test" mode. They will only work if the form is run using the Run button from the project pane.

### Adding Controls

Click on any of the Tools toolbars for Form windows to choose a control. Add the control to the form by clicking and dragging where the form should appear. Standard controls when added to a form will be preconfigured with default captions and styles.

We can choose the Text Box control from the Tools toolbar.

To add this control we click and drag where we would like it to appear.

The result is a text box control with default properties. We can change those properties by right clicking on the control and choosing Properties from the resultant context menu. For example, we might wish to change the Text property of the text box from "Text" to "Sample Text".

We can also choose the Check Box control and add it to the form by clicking and dragging.
The above illustration shows the form after we have added a check box and then changed the Caption property of the check box to "A Check Box".

**Selecting Controls**

Like many things in Manifold, we will often want to select controls so that we edit them. This section is a general introduction to selecting controls using regular Manifold selection commands. Many programmers will also use "smart mouse" selection commands as discussed later on in this topic.

To save space, the subsequent illustrations will not show the Form window border or the outer border of the form. When any controls are selected the form itself is deselected and the edit handles on the form's edges will disappear. To edit (resize) the form itself, deselect the controls and the form will automatically be selected for editing.

Controls appear in the form in their unselected forms.

We can select them using any of the usual Manifold Selection mouse moves with selection modes applied as in all other uses of selection. For example, drawing a box around them with Select Box will select the two controls above. The controls will be marked with selection borders having selection handles.

Whenever a selection is made in a form one of the controls is the primary selection and all of the other selected controls are in the secondary selection. The primary selection control is the one that is used to guide actions such as alignment. This is a concept widely used in professional development tools such as Visual Studio and has proven to be very fast and intuitive. When coupled with smart mouse selection modes it allows rapid manipulation of controls in forms.

Solid black edit handles indicate the primary selected control. The primary selection is the only control that can be resized with the cursor (choose another control as the primary to resize it). The primary control guides alignment and resizing of all other selected controls within alignment and resizing commands. The most recently selected control becomes the primary selected control.
When the mouse is committed to a selection command it works in selection mode only. When we unclick the selection control toolbar button the mouse becomes available for other activities, such as resizing or moving controls. If we **unclick** the Select Box tool used earlier we will be able to use the mouse again in the Form.

For example, we can click and drag (the mouse will change to the move cursor shown above) to move a control.

We have just clicked and dragged on one control and moved both controls to the left. We can also click and drag on an edit handle in the primary control to resize it. If we click and drag on the rightmost edit handle we can resize the text box’s width.

The result is a wider text box.

Note that all Manifold selection commands will work correctly with forms. For example, the **Edit menu’s Select All, Select None and Select Inverse** commands work as expected.

As seen in the illustration above, if one control in a form is selected, using **Select Inverse** will deselect it and will select the other controls.
Keyboard TAB Key Selection

Pressing the TAB key will cycle the selection through all of the controls in the form, selecting each in turn as the primary selected object. Press the TAB key to jump the selection from one control to another. At times this is a handy way of selecting controls within a crowded form or selecting controls that lie behind other controls.

Smart Mouse Selection

Manifold form windows support "smart mouse" selection like that used in Visual Basic forms editing. If the mouse is not occupied with a command mode (such as a regular selection command like Select Box) it may be used for smart mouse selection and editing as follows:

- **Click** Click on an unselected control to select it.
- **SHIFT Click** SHIFT Click on an unselected control to add it to the selection.
- **CTRL Click** CTRL Click on a selected control to make it the primary.
- **Click and Drag** Click and drag a primary selected control to move it. Clicking and dragging a secondary selected control when more than one control is selected will move all the controls.

The best way to learn the above is to create a form with some controls and to experiment. Smart mouse selection is a fast and intuitive way of editing forms.

Moving Controls

Controls in forms may be moved using either the mouse or keyboard keys. If several controls are selected, move logic is as follows:

- Clicking and dragging on the primary selected control will move only that control.
- Clicking and dragging on any of the other (secondary selected) controls will move all of the controls.
- Arrow keys on the keyboards will reposition all selected controls one pixel at a time. Use these to "nudge" controls into position.

Arrow Key Short Cuts to Reposition and Resize

When controls are selected, arrow keys on the keyboard use the following logic:

- **Repositioning**: Arrow keys on the keyboards will reposition all selected controls one pixel at a time. Use these to "nudge" controls into position.
- **Resizing**: When only one control is selected, a SHIFT-arrow key press will resize the control. The left and right arrows will move the right margin of the control to the right or to the left. The up and down arrows will move the lower margin of the control down or up.

Using Grids and Snap to Grid

Use View - Grid to turn on a grid in the form and then use Snap to Grid to constrain the mouse cursor to moving between grid points. This greatly assists in creating controls that are reasonably aligned.
By default, forms are shown in a form window without a grid.

Choose View - Grid to launch the Grid dialog. Check the Show grid box to turn on the grid. Grids are created on a 5-pixel spacing for forms by default.

The result is that a grid of dots appears every 5 pixels within the form.

Turn on Snap to Grid by pushing the Snap to Grid button or by choosing Edit - Snap to - Grid from the main menu. When Snap to Grid is on, the mouse cursor will move only between grid points on the form. Controls may therefore be created that are shaped exactly to locations marked by grid points.
This greatly facilitates drawing grid controls that are evenly aligned. Note that the grid is displayed only on the form itself and does not appear on the controls.

**Alignment Commands**

It's often the case that one's managers, clients and other demanding users will insist on well-organized dialogs where all the controls line up neatly. These are usually the same sorts of users who complain about spelling mistakes and who expect programs to come with documentation. Fortunately for professional programmers, Manifold's forms editor provides alignment commands so we can make pretty forms without stealing too much time from downloading mp3's.

The alignment controls are found on the Alignment toolbar.

- **Align Left** - Move objects so their left edges are aligned to the left edge of the primary selected object.
- **Align Top** - Move objects so their top edges are aligned to the top edge of the primary selected object.
- **Align Right** - Move objects so their right edges are aligned to the right edge of the primary selected object.
- **Align Bottom** - Move objects so their bottom edges are aligned to the bottom edge of the primary selected object.
- **Center Horizontally** - Move objects so that their centroid is centered horizontally in the form.
- **Center Vertically** - Move objects so that their centroid is centered vertically in the form.
- **Space Across** - Space objects evenly across the form in a horizontal direction.
- **Space Down** - Space objects evenly up and down the form in a vertical direction.
- **Same Width** - Resize objects so that the width of all is the same as the width of the primary selected object.
- **Same Width (Max)** - Resize objects so that the width of all is the same as the width of the widest object.
**Alignment Example**

To align the controls above, we begin by selecting them.

Press the **Align Left** command in the **Alignment toolbar**.

The controls will instantly align so that the left margin of each is lined up with the leftmost margin of the primary selected control. In this case, the primary control was the text box.
Clicking the Select None button in the selection toolbar to deselect all the controls shows us the new, improved, neat and clean controls arrangement. Excellent! ... Now we can get back to downloading mp3's...

Seeing Controls in their Run Time Form

Many controls in forms will change their appearance from design time to run time. Push the Test button (or right click on the form and choose Test from the context menu) to see the form in its run time appearance. The Test run will allow many controls to function visually as well, so tabs may be clicked or command buttons pressed to test the visual behavior of controls during run time. However, it is not the same as running the actual form / script combination. In particular, if a control is scripted to close the form it will not work from the Test environment. Therefore, if one does not equip the form with a Control Box “X” close button in the upper right corner there will be no obvious way to close the form when Test is run. In such cases, use ALT-F4 to close the form.

Tech Tips

Remember to unclick a control creation tool (such as Insert Check Box) before reaching into the form window to move things about. Otherwise, you’ll end up creating lots of little check boxes on top of each other when you think you are clicking and dragging controls about.

Remember to unclick a selection mode box (such as Select Box) before doing trying to move or resize. Otherwise, the move and resize cursors won’t appear because the system thinks you still want to select things.

If you edit a form’s properties to eliminate the System Menu or to make it borderless, you may end up with a form that provides no way to Close the form when it is tested. In such cases use ALT F4 to close the form since this keyboard shortcut to closing is always enabled.

Since the grid does not appear within controls, this effect can interfere with placing controls within a frame, since the interior of the frame will also be grid-free. Mouse action will still be aligned to the grid if Snap to Grid is on, but we won’t see the grid within the frame control. To see the grid when inserting controls within a frame, draw clusters of controls and then draw frames that contain them.

Suppose we’ve drawn two checkbox controls using the grid.

Using the Insert Frame control we draw a frame.
The frame appears on top of the checkbox controls. The checkbox controls cannot be seen because the frame’s interior is solid.

We push out the **Insert Frame** button to turn off that command and then click on the new frame to select it. We can then right click on the frame and choose **Order - Move to Bottom** to move the control below the checkboxes.

We can now see the checkbox controls. If desired, we can select them and move them about together for precise placement within the frame.

Most professional programmers work with a grid on and **Snap to Grid** enabled. Using grids together with smart mouse selection (with keyboard modifiers), use of keyboard arrow keys to move and resize controls and the **Alignment toolbar** controls results in fast and accurate form creation.
Forms and Scripts

Creating a form automatically creates an empty script that is associated with the form. The script refers to controls in the form to achieve the programmatic functioning desired for the form.

By default, the script uses VBScript language. Suppose we create a new form in the project. The minimum script created by default simply shows the form:

```vbnet
Sub Main
    Form.Visible = True
End Sub
```

We can write a more elaborate script that upon the press of a button displays a message box showing "Hello, World!" To do this, we would first add a command button to the form.

Using the Insert Command Button tool insert a command button into the form as described in the Editing Forms topic. Right click onto the new command button, choose Properties and in the Control tab change the default Name to Hello and in the General tab change the default Caption to Hello.

The form with the new command button is seen above in a form window open for editing. We can now click open the form's script and add code to handle a click of the Hello button. This will be an additional Sub:

```vbnet
Sub Main
    Form.Visible = True
End Sub

' -- Launch message box on click
Sub Hello_Click
    MsgBox "Hello, World!"
End Sub
```

If we run the form from the project pane it creates a form:

Every time we click the Hello button a message box will launch with our desired text:
We can click **OK** on the message box and then click **Hello** to display the message again as often as we like. To close the form we can click on the **X** box in the upper right corner. We can add a second button, a **Close** button that will close the form using our own command button.

To do so we use the Insert Command Button tool again to insert a second command button into the form. Right click onto the new command button, choose **Properties** and in the **Control** tab change the default **Name** to **Close** and in the **General** tab change the default **Caption** to **Close**.

After a little tinkering with the alignment commands as described in the Editing Forms topic we have two buttons that are the same size and neatly aligned.

In the form's script we add a **Sub** to handle clicks of the **Close** button:

```vbnet
Sub Main
    Form.Visible = True
End Sub

' -- Launch message box on click
Sub Hello_Click
    MsgBox "Hello, World!"
End Sub

' -- close form
Sub Close_Click
    Form.Visible = False
End Sub
```

Running the form now shows it has two command buttons. The **Hello** button pops open a message box and the **Close** button closes the form.

A more sophisticated form would use controls like text boxes to display text that is generated programmatically. For example, we might create a form to view the contents of the **Employees** table of the **Nwind.mdb** sample database:
The script that powers this form is directly analogous to the simple example with two buttons. The script launches with a Main routine that makes the form visible. A Close button closes the form. Routines for each button apply some program logic and call the Reload routine to redisplay the form. The Reload routine loads up the various text boxes with appropriate text strings taken from the table record.

' -- launch form on startup
Sub Main
    Form.Visible = True
End Sub

' ---------------------------
Dim Employees
EmployeeIndex = -1
EmployeeCount = 10

' ---------------------------

' -- jump to first employee on load
Sub Form_OnLoad

    ' -- set up global variables
    Set Components = Application.ActiveDocument.ComponentSet
    EmpPos = Components.ItemByName("Employees")
    If EmpPos < 0 Then
        MsgBox ":'Employees' table not found."
        Form.Visible = False
    End If
    Set Employees = Components(EmpPos)
    EmployeeIndex = 0
    EmployeeCount = Employees.RecordSet.Count

    ' -- load controls
    Reload

End Sub

Notes: Education includes a BA in Psychology from Colorado State University in 1970. She also completed "The Art of the Cold Call." Nancy is a member of Toastmasters International.
End Sub

' -- close form
Sub Close_Click
    Form.Visible = False
End Sub

' -- jump to first employee
Sub First_Click
    If EmployeeIndex <> 0 Then
        EmployeeIndex = 0
        Reload
    End If
End Sub

' -- jump to last employee
Sub Last_Click
    If EmployeeIndex <> EmployeeCount-1 Then
        EmployeeIndex = EmployeeCount-1
        Reload
    End If
End Sub

' -- jump to next employee if any
Sub Next_Click
    If EmployeeIndex < EmployeeCount-1 Then
        EmployeeIndex = EmployeeIndex+1
        Reload
    End If
End Sub

' -- jump to previous employee if any
Sub Previous_Click
    If EmployeeIndex > 0 Then
        EmployeeIndex = EmployeeIndex-1
        Reload
    End If
End Sub

' -- load current employee
Sub Reload
    Set Rec = Employees.RecordSet(EmployeeIndex)

    ' -- set employee readouts
    Employee.Text = Rec.Data("First Name") & " " & Rec.Data("Last Name") & " (#" & CStr(EmployeeIndex+1) & ")"]
    Title.Text = Rec.Data("Title")
    BirthDate.Text = Rec.Data("Birth Date")
    HireDate.Text = Rec.Data("Hire Date")
    Country.Text = Rec.Data("Country")
    Region.Text = Rec.Data("Region")
    City.Text = Rec.Data("City")
    PostalCode.Text = Rec.Data("Postal Code")
Address.Text = Rec.Data("Address")
Notes.Text = Rec.Data("Notes")
End Sub

We've named the text box controls the same as the text captions (which are separate controls), so that code like...

Region.Text = Rec.Data("Region")

... refers to the text box named Region.

The form shown above allows us to step through the Employees table using First, Previous, Next and Last buttons. With each press of a button the appropriate record info will be displayed in the text boxes. The form is intended as a simple example but even so it is quite useful.

We could extend the form above by adding one more button, a Select button to select the record currently shown in the table. Just like adding buttons in the two button example at the beginning of this topic, we use the Insert Command Button tool to draw a new command button in the form. Right click onto the new command button, choose Properties and in the Control tab change the default Name to Select and in the General tab change the default Caption to Select. Use the alignment tools to make the button the same size as the other buttons and aligned neatly in the same row. We've placed the new Select button in the lower left corner of the form.

We can then add one more Sub to the script to handle any clicks of the Select button:

Sub Select_Click
' clear any selection
Set records = Employees.RecordSet
For nItem = 0 To records.Count-1
    Set record = records(nItem)
    record.Mask = record.Mask And Not 1
Next
' create selection
Set record = Employees.RecordSet(EmployeeIndex)
record.Mask = record.Mask Or 1
End Sub

This routine first clears the selection and then selects whatever is the current employee seen in the form.

Using Languages other than VBScript

Form scripts can use languages other than VBScript, for example, Javascript. The syntax of event handlers will vary from language to language.

Here is the Javascript version of the script for the form discussed at the beginning of this topic:

function Main() {
    Form.Visible = true;
}

function Hello::Click() {


Note that form scripts cannot use .NET languages such as C# or VB .NET. This is only a minor issue since .NET scripts can create forms in code, without using form components, through the use of Windows Forms. Advanced script writers will also frequently design forms outside of Manifold using Visual Studio .NET or other tools, and call these forms from Manifold scripts.

Frequently Used Controls

Control - Tools Toolbar

The Tools toolbar when used with forms hosts the most frequently used controls in Forms. There are many more controls available in the advanced tools toolbar. For advanced controls, see the Tools (Advanced) toolbar topic.

Click on a control in the Tools toolbar and then click and drag in the Form window to insert the control in the form in the position and size indicated by the mouse click and drag. Forms are then customized using the properties for each control.

Some attributes of controls, such as their size or position are properties not of the control but of the container for the control, in this case the Form itself. To change the size or position of controls we can simply click on them and resize or drag them to a new position.

Properties normally both return and set that particular property. The easiest way to edit properties is through the context menu's properties dialog for that control.

See the individual controls topics for details on each control.

**Insert Check Box** - Inserts check box controls.

**Insert Option Button** - Inserts option button controls. Only one option button in a group may be selected.

**Insert Command Button** - Inserts command button controls.

**Insert Frame** - Draws frame with caption. Frames are used to set off groups of controls, often option buttons.

**Insert Text Box** - Inserts text box. Also called an edit field or edit control. This control can display text entered by the programmer at design time or entered by the user or assigned to the control by other code at run time.

**Insert List Box** - Inserts list box controls.

**Insert Combo Box** - Inserts combo box controls.

**Insert Horizontal Scroll Bar** - Inserts horizontal scroll bar controls.

**Insert Vertical Scroll Bar** - Inserts vertical scroll bar controls.

Lightweight Controls
Manifold System implements the above controls using Microsoft's "windowless" lightweight ActiveX controls found in the Microsoft MSWLess.ocx file. Lightweight controls do not have an hWnd property, which dramatically conserves computer resources and enables rapid launch. Also, Dynamic Data Exchange (DDE) is not supported within lightweight controls. In all other aspects the lightweight version of the control works exactly the same as the standard VB version. There is a simple "group" substitute in the one case where the windowless nature of the controls has an effect, and there are many other choices other than DDE within Microsoft data access methodology. For these reasons, use of lightweight controls has no negative impact in scripts.

Microsoft's MSWLess.ocx ActiveX controls used within Manifold are **Unicode-enabled**. Following standard Windows practise, no conversion of Unicode data occurs when used within a Unicode-aware operating system such as Microsoft Windows Server 2003, Windows XP or Microsoft Windows 2000. The control will pass data with no conversion involved in such systems. When used within non-Unicode-enabled operating Windows editions (so long as such may be supported), data will be converted from ANSI to Unicode and back.

**Note:** All properties and controls from the lightweight series are prefaced with "wl" to indicate they are the windowless lightweight versions of the controls and properties settings. The "wl" prefix is suppressed in the documentation below to enhance legibility.

**Using the Group Property**

For the most part, the windowless nature of Microsoft lightweight controls does not interfere with the creation of whatever form we like. The key exception is the use of frames to group option buttons. In VB, frames are windows and will automatically group controls, such as option buttons, within them. This lets the system automatically require that only one option button within the same frame be chosen. It is standard practice within VB to segregate OptionButton controls within their own Frame control.

With windowless controls, we need to set the group property of each object that is to be in a particular group to the same unique string. For example, we might draw a frame called Truck Display Options that included three option buttons labeled Ahead, On Time and Delayed. Assuming we named our option buttons in a way that is easy to remember, we could assign each of these three option buttons to the same group with:

```vbnet
Ahead.Group = "TruckDisplayOption"
OnTime.Group = "TruckDisplayOption"
Delayed.Group = "TruckDisplayOption"
```

The string can be whatever we want it to be as long as it is unique for each group desired. It's usually wise to use a string that is self-documenting.

The above is not necessary with VB, but then again in VB it is really easy to not create controls in a form in the right order and so end up thinking that a control is in a frame when it is not (it's just positioned above the frame). Declaring the group property for each option button in the above explicit fashion eliminates such possible errors.

**Other Controls**

The controls in the Tools (Advanced) toolbar as well as all other controls in .ocx files provided with manifold are implemented using standard Microsoft window controls having an hWnd property.

**Form Properties for Controls**

Manifold adds an extra tab, the Control tab, to the usual tabs that appear in the design time properties dialog for controls. This tab provides access to the properties for this control maintained by the form. This extra tab will appear at design time for all controls. Although strictly speaking these are properties of the form that is the container for the control and not of the control, they are easiest to specify at design time if they appear within each control's properties dialog.

**Control**

- **Name**  The name used by the form to identify this control.
- **Tag**    A string available for free use by the programmer. Often used for comments.
Left  Left margin position of the control.
Top   Top margin position of the control.
Width Width of the control.
Height Height of the control.
Visible Make this control visible or invisible.
Control - Check Box

Insert a check box into the form.

To Add a Check Box to a Form

1. Click on the Insert Check Box button in the Tools toolbar.
2. Click and drag the location and size where it is to be created in the form.
3. Continue to create any additional check boxes desired.
4. Right click onto each check box and choose Properties from the context menu.
5. Set the properties for the check box as desired.
6. Move, resize and align the check box as desired.

Properties normally both return and set that particular property. The easiest way to edit properties is through the context menu's properties dialog for that control.

Some attributes of controls, such as their size or position are properties not of the control but of the container for the control, in this case the Form itself. To change the size or position of controls we can simply click on them and resize or drag them to a new position.

Keep in mind that programs can change the properties of controls at run time. Although this documentation is written as though the properties are set at design time using the Properties dialog for this control, there are many situations in which program code will be used to change a control's properties at run time.

Note: All properties and controls are prefaced with "wl" to indicate they are the windowless lightweight versions of the controls and properties settings. The "wl" prefix is suppressed in the documentation below to enhance legibility.

Properties

In addition to the properties listed below, a Control tab provides access to certain frequently used form properties for this control. See the Form Properties topic for information on those properties.

Check Box

General

Caption  Text used to label the check box. Font is set in the font tab.

MousePointer  The type of mouse pointer displayed when over the check box. Choose from 16 different types plus a custom pointer.

Appearance  3D or Flat. 3D is the standard Windows look.

Alignment  Places check box to left or right of caption within the size of the control.

OLEDropMode  Set to None (default) if the check box does not accept OLE drops and is to display the No Drop cursor if an OLE drop is attempted. Set to Manual, the check box will trigger OLE drop events, allowing programs to handle the OLE drop operation in code.

Style  Standard (default) or Graphical. In Standard style check boxes display as a box/button with the caption label next to it. In graphical style, check boxes display as a button that can be pushed in/out and may also display an associated graphic.

Value  Default state of the control: Unchecked, Checked and Grayed (dimmed).
**Enabled**
True or False. True means the object can respond to user-generated events, false prevents it from responding. We can disable a control like a text box if we wish it to simply display information in a read-only way.

**UseMaskColor**
If checked, enables use of mask color in a button's picture if the **Style** is set to graphical. A mask color is the color that is to be made transparent.

### Color

**Back Color**
Color to be used for the check box's background, normally seen only when Flat **Style** is used.

**Fore Color**
Color to be used for the check box's foreground, the color of the text caption.

**Mask Color**
Color to be treated as transparent in an image used with the button when **UseMaskColor** is checked and **Style** is set to graphical.

**Color Set**
Choose from Standard Colors (standard Windows non-dithered colors) or Windows System Colors. The latter will be defined by the user's Control Panel settings and is normally the setting used so that the form changes appearance like the rest of Windows if the Control Panel settings are changed.

**Color Palette**
Displays available colors. Click on the property to be changed to highlight it in the **Properties** pane, click on the desired color in the **Color Palette** pane and then press **Apply**.

**Edit Custom Color**
Change the custom color presented in the **Color Palette** when the **Color Set** is set to Windows System Colors.

### Font

**Properties**
Font properties that may be changes.

**Font**
Choose a font installed on this system. It's wise to choose standard Windows fonts such as **MS Sans Serif** that are universally available.

**Size**
Size of font, in points.

**Effects**
Bold, Italic, Underline or Strikeout.

**Sample Text**
A preview of the selected settings.

### Picture

**DisabledPicture**
Graphic to show when control is disabled if **Style** is set to graphical.

**DownPicture**
Graphic to show on an enabled control in the DOWN position if **Style** is set to graphical.

**MouseIcon**
Custom icon to use when the **MousePointer** property is set to 99. Like Visual Basic, will not load animated cursor (**.ani**) files.

**Picture**
Graphic to show on an enabled control in the UP position if **Style** is set to graphical.
Note: graphics are loaded at design time and saved within the project. When browsing to find the graphic to be used, .bmp, .ico, .wmf, .gif or .jpeg files may be used.

Examples

Check boxes are often used as "mode" buttons in graphical style. Suppose we want our form to control the display of a map that shows airports, bus stations and train stations. Using Manifold's image capabilities, we have created six small images, each of which is 30 x 30 pixels in size:

The gray versions are simply copies of the green versions that were desaturated and had the lightness increased using Manifold's Hue / Saturation command. We first create three check boxes and clear their caption property:

In the default standard style they are just ordinary check boxes. We have created these check boxes with a Height and a Width of 40 pixels each.

If we choose graphical style for each of the check boxes, they appear as a 3D button.
We can also choose the **Picture** property for each of them to use the gray picture in their "up" position.

For the **DownPicture** property we use the green picture.

When the form is run, pressing the button (equivalent to checking the check box) switches the green picture into the button. This is a nice way of emphasizing that this button is pushed in. In the illustration above we have chosen airports and bus stations.
Here, we have chosen just train stations.

**Note:** It's very cool we can use graphics so easily in Manifold forms. However, one should use them with restraint to avoid cluttering up forms with distracting graphical junk. Sometimes a plain check box is a better approach.
Control - Option Button

Option buttons (also called radio buttons in programmer slang) are used for choices where only one choice in a group is allowed.

To Add an Option Button to a Form

1. Click on the Insert Option Button button in the Tools toolbar.
2. Click and drag the location and size where it is to be created in the form.
3. Continue to create any additional option buttons desired.
4. Right click onto each button and choose Properties from the context menu.
5. Set the properties for the option button as desired. Note that each option button must belong to a group.
6. Move, resize and align the option button as desired.

Properties normally both return and set that particular property. The easiest way to edit properties is through the context menu’s properties dialog for that control.

Some attributes of controls, such as their size or position are properties not of the control but of the container for the control, in this case the Form itself. To change the size or position of controls we can simply click on them and resize or drag them to a new position.

Keep in mind that programs can change the properties of controls at run time. Although this documentation is written as though the properties are set at design time using the Properties dialog for this control, there are many situations in which program code will be used to change a control's properties at run time.

Note: All properties and controls are prefaced with "wl" to indicate they are the windowless lightweight versions of the controls and properties settings. The "wl" prefix is suppressed in the documentation below to enhance legibility.

Using the Group Property

For the most part, the windowless nature of Microsoft lightweight controls does not interfere with the creation of whatever form we like. The key exception is the use of frames to group option buttons. In VB, frames are windows and will automatically group controls, such as option buttons, within them. This lets the system automatically require that only one option button within the same frame be chosen. It is standard practice within VB to segregate OptionButton controls within their own Frame control.

With windowless controls, we need to set the group property of each object that is to be in a particular group to the same unique string. For example, we might draw a frame called Truck Display Options that included three option buttons labeled Ahead, On Time and Delayed. Assuming we named our option buttons in a way that is easy to remember, we could assign each of these three option buttons to the same group with:

```
Ahead.Group = "TruckDisplayOption"
OnTime.Group = "TruckDisplayOption"
Delayed.Group = "TruckDisplayOption"
```

The string can be whatever we want it to be as long as it is unique for each group desired. It's usually wise to use a string that is self-documenting.

Default Values for Option Buttons

One and only one option button in a group should have a True value at all times. Because option buttons are created by default with a False value, after creating all of the option buttons in a group and setting their Group property to the same string, one of the buttons should have its Value box checked True so that it is by default the ON button.

Properties
In addition to the properties listed below, a Control tab provides access to certain frequently used form properties for this control. See the Form Properties topic for information on those properties.

Option Button

**General**

**Caption** Text used to label the option button. Font is set in the font tab.

**Group** A text string. All option buttons with the same text string are considered part of the same group. Checking one option button in a group will uncheck all other option buttons in that group.

**MousePointer** The type of mouse pointer displayed when over the option button. Choose from 16 different types plus a custom pointer.

**Appearance** 3D or Flat. 3D is the standard Windows look.

**Alignment** Places option button to left or right of caption within the size of the control.

**OLEDropMode** Set to None (default) if the option button does not accept OLE drops and is to display the No Drop cursor if an OLE drop is attempted. Set to Manual, the option button will trigger OLE drop events, allowing programs to handle the OLE drop operation in code.

**Style** Standard (default) or Graphical. In Standard style option buttons display as a box/button with the caption label next to it. In graphical style, option buttons display as a button that can be pushed in/out, and may also display an associated graphic.

**Value** Default state of the control: Unchecked, Checked and Grayed (dimmed). Only one option button in a group can be checked.

**Enabled** True or False. True means the object can respond to user-generated events, false prevents it from responding. We can disable a control like a text box if we wish it to simply display information in a read-only way.

**UseMaskColor** If checked, enables use of mask color in a button’s picture if the Style is set to graphical. A mask color is the color that is to be made transparent.

**Color**

**Back Color** Color to be used for the option button’s background, normally seen only when Flat Style is used.

**Fore Color** Color to be used for the option button’s foreground, the color of the text caption.

**Mask Color** Color to be treated as transparent in an image used with the button when UseMaskColor is checked and Style is set to graphical.

**Color Set** Choose from Standard Colors (standard Windows non-dithered colors) or Windows System Colors. The latter will be defined by the user's Control Panel settings and is normally the setting used so that the form changes appearance like the rest of Windows if the Control Panel settings are changed.

**Color Palette** Displays available colors. Click on the property to be changed to highlight it in the Properties pane, click on the desired color in the Color Palette pane and then
Edit Custom Color

Change the custom color presented in the Color Palette when the Color Set is set to Windows System Colors.

Font

Properties

Font properties that may be changes.

Font

Choose a font installed on this system. It's wise to choose standard Windows fonts such as MS Sans Serif that are universally available.

Size

Size of font, in points.

Effects

Bold, Italic, Underline or Strikeout.

Sample Text

A preview of the selected settings.

Picture

DisabledPicture

Graphic to show when control is disabled if Style is set to graphical.

DownPicture

Graphic to show on an enabled control in the DOWN position if Style is set to graphical.

MouseIcon

Custom icon to use when the MousePointer property is set to 99. Like Visual Basic, will not load animated cursor (.ani) files.

Picture

Graphic to show on an enabled control in the UP position if Style is set to graphical.

Note: graphics are loaded at design time and saved within the project. When browsing to find the graphic to be used, .bmp, .ico, .wmf, .gif or .jpeg files may be used.

Examples

We will use a graphical style for this particular option button and also add a picture.

Default option button with standard Style.

The same button with graphical Style. The size of the 3D graphical button is set by the size of the control as shown in the form editor when it is selected.
To add a picture to this control, we use the Properties dialog's Picture tab as shown above.

The result is a picture. To get rid of the white background, we can check UseMaskColor and set the mask color to white.

That's better!

Note: To create this example, we used one of the standard manifold.net images from our corporate clip-art collection. The particular picture used had been optimized for use on web sites and so includes some anti-aliasing pixels in lighter colors on the edges of the red and yellow triangles. These anti-aliasing pixels are lighter shades of color between pure white and red. Using a pure white mask color to get rid of the background will leave the anti-aliasing pixels in place to provide a slight "halo" of lighter pixels that are visible in some spots against the gray color of the button. We can eliminate this effect by using an image that does not include anti-aliasing pixels in the first place.
**Control - Command Button**

Command buttons are used to issue commands in a form. The **OK** and **Cancel** buttons often seen in Windows dialogs are command buttons, for example.

**To Add a Command button to a Form**

1. Click on the **Insert Command button** button in the **Tools toolbar**.
2. Click and drag the location and size where it is to be created in the form.
3. Continue to create any additional command buttons desired.
4. Right click onto each command button and choose **Properties** from the context menu.
5. Set the properties for the command button as desired.
6. Move, resize and align the command button as desired.

Properties normally both return and set that particular property. The easiest way to edit properties is through the context menu’s **properties** dialog for that control.

Some attributes of controls, such as their size or position are properties not of the control but of the container for the control, in this case the Form itself. To change the size or position of controls we can simply click on them and resize or drag them to a new position.

Keep in mind that programs can change the properties of controls at run time. Although this documentation is written as though the properties are set at design time using the **Properties** dialog for this control, there are many situations in which program code will be used to change a control’s properties at run time.

**Note:** All properties and controls are prefaced with "wl" to indicate they are the windowless lightweight versions of the controls and properties settings. The "wl" prefix is suppressed in the documentation below to enhance legibility.

**Properties**

In addition to the properties listed below, a **Control tab** provides access to certain frequently used form properties for this control. See the Form Properties topic for information on those properties.

**Command Button**

**General**

- **Caption** Text used to label the command button. Font is set in the font tab.
- **MousePointer** The type of mouse pointer displayed when over the command button. Choose from 16 different types plus a custom pointer.
- **Appearance** 3D or Flat. 3D is the standard Windows look. Flat has no effect with command buttons.
- **OLEDropMode** Set to None (default) if the command button does not accept OLE drops and is to display the No Drop cursor if an OLE drop is attempted. Set to Manual, the command button will trigger OLE drop events, allowing programs to handle the OLE drop operation in code.
- **Style** Standard (default) or Graphical. In standard style command buttons display as a button with the caption on top. In graphical style, command buttons may also display an associated graphic.
- **Enabled** True or False. True means the object can respond to user-generated events, false prevents it from responding. We can disable a control like a text box if we wish it to simply display information in a read-only way.
UseMaskColor  If checked, enables use of mask color in a button’s picture if the Style is set to graphical. A mask color is the color that is to be made transparent.

Color

Back Color  Color to be used for the command button’s background, normally seen only when Flat Style is used.

Fore Color  Color to be used for the command button’s foreground, the color of the text caption.

Mask Color  Color to be treated as transparent in an image used with the button when UseMaskColor is checked and Style is set to graphical.

Color Set  Choose from Standard Colors (standard Windows non-dithered colors) or Windows System Colors. The latter will be defined by the user’s Control Panel settings and is normally the setting used so that the form changes appearance like the rest of Windows if the Control Panel settings are changed.

Color Palette  Displays available colors. Click on the property to be changed to highlight it in the Properties pane, click on the desired color in the Color Palette pane and then press Apply.

Edit Custom Color  Change the custom color presented in the Color Palette when the Color Set is set to Windows System Colors.

Font

Properties  Font properties that may be changes.

Font  Choose a font installed on this system. It’s wise to choose standard Windows fonts such as MS Sans Serif that are universally available.

Size  Size of font, in points.

Effects  Bold, Italic, Underline or Strikeout.

Sample Text  A preview of the selected settings.

Picture

DisabledPicture  Graphic to show when control is disabled if Style is set to graphical.

DownPicture  Graphic to show on an enabled control in the DOWN position if Style is set to graphical.

MouseIcon  Custom icon to use when the MousePointer property is set to 99. Like Visual Basic, will not load animated cursor (.ani) files.

Picture  Graphic to show on an enabled control in the UP position if Style is set to graphical.

Note: graphics are loaded at design time and saved within the project. When browsing to find the graphic to be used, .bmp, .ico, .wmf, .gif or .jpeg files may be used.
Control - Frame

Frames are used to provide a border within the form that includes a caption. Frames are used to organize forms.

Frames are also used to organize option buttons into related groups. In compiled Visual Basic, frames will automatically group option buttons if the option buttons and frames are created in the correct sequence. In Manifold, frames are simply visual artifacts used to organize the form. We can use the group property of option buttons (see below) to organize them in a standard VB way.

To Add an Frame to a Form

1. Click on the Insert Frame button in the Tools toolbar.
2. Click and drag the location and size where it is to be created in the form.
3. Continue to create any additional frames desired.
4. Right click onto each button and choose Properties from the context menu.
5. Set the properties for the frame as desired. Note that each frame must belong to a group.
6. Move, resize and align the frame as desired.

Properties normally both return and set that particular property. The easiest way to edit properties is through the context menu's properties dialog for that control.

Some attributes of controls, such as their size or position are properties not of the control but of the container for the control, in this case the Form itself. To change the size or position of controls we can simply click on them and resize or drag them to a new position.

Keep in mind that programs can change the properties of controls at run time. Although this documentation is written as though the properties are set at design time using the Properties dialog for this control, there are many situations in which program code will be used to change a control's properties at run time.

Note: All properties and controls are prefaced with "wl" to indicate they are the windowless lightweight versions of the controls and properties settings. The "wl" prefix is suppressed in the documentation below to enhance legibility.

Using the Group Property

For the most part, the windowless nature of Microsoft lightweight controls does not interfere with the creation of whatever form we like. The key exception is the use of frames to group option buttons. In VB, frames are windows and will automatically group controls, such as option buttons, within them. This lets the system automatically require that only one option button within the same frame be chosen. It is standard practice within VB to segregate OptionButton controls within their own Frame control.

With windowless controls such as the frames control used within Manifold, we need to set the group property of each object that is to be in a particular group to the same unique string. For example, we might draw a frame called Truck Display Options that included three option buttons labeled Ahead, On Time and Delayed. Assuming we named our option buttons in a way that is easy to remember, we could assign each of these three option buttons to the same group with:

```vbnet
Ahead.Group = "TruckDisplayOption"
OnTime.Group = "TruckDisplayOption"
Delayed.Group = "TruckDisplayOption"
```

The string can be whatever we want it to be as long as it is unique for each group desired. It's usually wise to use a string that is self-documenting.

Properties

In addition to the properties listed below, a Control tab provides access to certain frequently used form properties for this control. See the Form Properties topic for information on those properties.
Frame

General

Caption  Text used to label the frame. Font is set in the font tab.

MousePointer  The type of mouse pointer displayed when over the frame. Choose from 16 different types plus a custom pointer

Appearance  3D or Flat. 3D is the standard Windows look. Flat results in a rectangular square of background color with a foreground color border and caption.

BorderStyle  FixedSingle (default) or None. None may be used with Flat Appearance to achieve an uncaptioned rectangle of background color.

OLEDropMode  Set to None (default) if the frame does not accept OLE drops and is to display the No Drop cursor if an OLE drop is attempted. Set to Manual, the frame will trigger OLE drop events, allowing programs to handle the OLE drop operation in code.

Enabled  True or False. True means the object can respond to user-generated events, false prevents it from responding. We can disable a control like a text box if we wish it to simply display information in a read-only way.

Color

Back Color  Color to be used for the frame's background, normally seen only when Flat Style is used.

Fore Color  Color to be used for the frame's foreground, the color of the text caption.

Color Set  Choose from Standard Colors (standard Windows non-dithered colors) or Windows System Colors. The latter will be defined by the user's Control Panel settings and is normally the setting used so that the form changes appearance like the rest of Windows if the Control Panel settings are changed.

Color Palette  Displays available colors. Click on the property to be changed to highlight it in the Properties pane, click on the desired color in the Color Palette pane and then press Apply.

Edit Custom Color  Change the custom color presented in the Color Palette when the Color Set is set to Windows System Colors.

Font

Properties  Font properties that may be changes.

Font  Choose a font installed on this system. It's wise to choose standard Windows fonts such as MS Sans Serif that are universally available.

Size  Size of font, in points.

Effects  Bold, Italic, Underline or Strikeout.

Sample Text  A preview of the selected settings.
Examples

- Truck Display Options
  - Ahead
  - On Time
  - Delayed
  - Update
Control - Text Box

Text boxes are used to display text and to accept text entry from users. They may be used either dynamically to display text from various data sources or (something of a kludge, but very handy) as a means of creating fixed labels within forms, built-in documentation or other fixed text.

At run time, Windows will provide text boxes with the full set of keyboard and text navigation controls normally associated with text in Windows.

To Add a Text Box to a Form

1. Click on the Insert Text box button in the Tools toolbar.
2. Click and drag the location and size where it is to be created in the form.
3. Continue to create any additional text boxes desired.
4. Right click onto each text box and choose Properties from the context menu.
5. Set the properties for the text box as desired.
6. Move, resize and align the text box as desired.

Properties normally both return and set that particular property. The easiest way to edit properties is through the context menu's properties dialog for that control.

Some attributes of controls, such as their size or position are properties not of the control but of the container for the control, in this case the Form itself. To change the size or position of controls we can simply click on them and resize or drag them to a new position.

Keep in mind that programs can change the properties of controls at run time. Although this documentation is written as though the properties are set at design time using the Properties dialog for this control, there are many situations in which program code will be used to change a control's properties at run time.

Note: All properties and controls are prefaced with "wl" to indicate they are the windowless lightweight versions of the controls and properties settings. The "wl" prefix is suppressed in the documentation below to enhance legibility.

Properties

In addition to the properties listed below, a Control tab provides access to certain frequently used form properties for this control. See the Form Properties topic for information on those properties.

Text box

General

MousePointer The type of mouse pointer displayed when over the text box. Choose from 16 different types plus a custom pointer
Alignment  Left justify, right justify, or center the text in the box.
Appearance  3D or Flat. 3D is the standard Windows look. Flat results in a rectangular square of background color with a foreground color border.
BorderStyle  FixedSingle (default) or None. None may be used with Flat Appearance to achieve an uncaptioned rectangle of background color.
OLEDragMode  DragManual (default) or DragAutomatic. DragManual means the program handles all OLE drag/drop operations, while DragAutomatic means the text box handles all OLE drag/drop operations.
OLEDropMode  Set to None (default) if the text box does not accept OLE drops and is to display the No Drop cursor if an OLE drop is attempted. Set to Manual, the text box will trigger OLE
Programming

don't events, allowing programs to handle the OLE drop
operation in code.

**ScrollBars**
NoScrollbars, Horizontal, Vertical or Both. Set as desired
to cause scroll bars to appear. Scroll bars appear only if
**MultiLine** is set True (checked). Scroll bars will always
appear in text boxes if enabled by this property even if the
text does not extend beyond the borders of the box.

**Enabled**
True or False. True means the object can respond to
user-generated events, false prevents it from responding.
We can disable a control like a text box if we wish it to
simply display information in a read-only way.

**HideSelection**
True or False. True means that highlighted text will not
appear highlighted when the text box loses the focus.
False means that highlighted text stays highlighted even
when another form or dialog has the focus.

**Locked**
True or False. True means the user can scroll and
highlight the text in the box but cannot change it. The
program can change the text by modifying the text box's
**Text** property. False (default) means users can edit the
text in the text box.

**MultiLine**
True or False. True (checked) enables multiple lines of
text. A multiple line text box will wrap text as the user
enters text that extends past the text box. If a horizontal
scroll bar is present text will not wrap. On a form with no
default button, pressing ENTER in a multiple line text box
will move the focus to the next line. If a default button
exists in the form, the user must press CTRL Enter to
move to the next line.

**PasswordChar**
Used to create password boxes in dialogs. Placing any
character or string in this box sets that character (or,
the first character in the field) as the character to use for
whatever is typed into that box. Virtually all applications
use the asterisk character. If the **MultiLine** property is
set True, setting the **PasswordChar** property will have no
effect.

**MaxLength**
An integer that limits the number of characters that can
be entered into the text box. Default is 0, meaning no
limit. Text exceeding the given value in length will have
extra characters truncated.

**Text**
**Text** Text contents for this text box.

**Color**

**Back Color**
Color to be used for the text box's background.

**Fore Color**
Color to be used for the text box's foreground, the color of
the text caption.

**Color Set**
Choose from Standard Colors (standard Windows non-
dithered colors) or Windows System Colors. The latter
will be defined by the user's Control Panel settings and is
normally the setting used so that the form changes
appearance like the rest of Windows if the Control Panel
settings are changed.

**Color Palette**
Displays available colors. Click on the property to be
changed to highlight it in the **Properties** pane, click on
the desired color in the **Color Palette** pane and then
press **Apply**.
### Edit Custom Color
Change the custom color presented in the **Color Palette** when the **Color Set** is set to Windows System Colors.

### Font Properties
Font properties that may be changed.

- **Font**
  Choose a font installed on this system. It's wise to choose standard Windows fonts such as **MS Sans Serif** that are universally available.

- **Size**
  Size of font, in points.

- **Effects**
  Bold, Italic, Underline or Strikeout.

- **Sample Text**
  A preview of the selected settings.
Control - Static Text Box

Static text boxes are used to display text not modified by users. Static text boxes are a variation of ordinary, editable text boxes.

To Add a Text Box to a Form

1. Click on the Insert Static Text Box button in the Tools toolbar.
2. Click and drag the location and size where it is to be created in the form.
3. Continue to create any additional text boxes desired.
4. Right click onto each text box and choose Properties from the context menu.
5. Set the properties for the text box as desired.
6. Move, resize and align the text box as desired.

Properties normally both return and set that particular property. The easiest way to edit properties is through the context menu’s properties dialog for that control.

Some attributes of controls, such as their size or position are properties not of the control but of the container for the control, in this case the Form itself. To change the size or position of controls we can simply click on them and resize or drag them to a new position.

Keep in mind that programs can change the properties of controls at run time. Although this documentation is written as though the properties are set at design time using the Properties dialog for this control, there are many situations in which program code will be used to change a control’s properties at run time.

**Note:** All properties and controls are prefaced with "wl" to indicate they are the windowless lightweight versions of the controls and properties settings. The "wl" prefix is suppressed in the documentation below to enhance legibility.

### Properties

In addition to the properties listed below, a Control tab provides access to certain frequently used form properties for this control. See the Form Properties topic for information on those properties.

#### Text box

##### General

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MousePointer</td>
<td>The type of mouse pointer displayed when over the text box. Choose from 16 different types plus a custom pointer.</td>
</tr>
<tr>
<td>Alignment</td>
<td>Left justify, right justify, or center the text in the box.</td>
</tr>
<tr>
<td>Appearance</td>
<td>3D or Flat. 3D is the standard Windows look. Flat results in a rectangular square of background color with a foreground color border.</td>
</tr>
<tr>
<td>BorderStyle</td>
<td>FixedSingle (default) or None. None may be used with Flat Appearance to achieve an uncaptioned rectangle of background color.</td>
</tr>
<tr>
<td>OLEDragMode</td>
<td>DragManual (default) or DragAutomatic. DragManual means the program handles all OLE drag/drop operations, while DragAutomatic means the text box handles all OLE drag/drop operations.</td>
</tr>
<tr>
<td>OLEDropMode</td>
<td>Set to None (default) if the text box does not accept OLE drops and is to display the No Drop cursor if an OLE drop is attempted. Set to Manual, the text box will trigger OLE drop events, allowing programs to handle the OLE drop operation in code.</td>
</tr>
<tr>
<td>ScrollBars</td>
<td>NoScrollbars, Horizontal, Vertical or Both. Set as desired.</td>
</tr>
</tbody>
</table>
to cause scroll bars to appear. Scroll bars appear only if `MultiLine` is set True (checked). Scroll bars will always appear in text boxes if enabled by this property even if the text does not extend beyond the borders of the box.

**Enabled**  
True or False. True means the object can respond to user-generated events, false prevents it from responding. We can disable a control like a text box if we wish it to simply display information in a read-only way.

**HideSelection**  
True or False. True means that highlighted text will not appear highlighted when the text box loses the focus. False means that highlighted text stays highlighted even when another form or dialog has the focus.

**Locked**  
True or False. True means the user can scroll and highlight the text in the box but cannot change it. The program can change the text by modifying the text box's `Text` property. False (default) means users can edit the text in the text box.

**MultiLine**  
True or False. True (checked) enables multiple lines of text. A multiple line text box will wrap text as the user enters text that extends past the text box. If a horizontal scroll bar is present text will not wrap. On a form with no default button, pressing ENTER in a multiple line text box will move the focus to the next line. If a default button exists in the form, the user must press CTRL Enter to move to the next line.

**PasswordChar**  
Used to create password boxes in dialogs. Placing any character or string in this box sets that that character (or, the first character in the field) as the character to use for whatever is typed into that box. Virtually all applications use the asterisk character. If the `MultiLine` property is set True, setting the `PasswordChar` property will have no effect.

**MaxLength**  
An integer that limits the number of characters that can be entered into the text box. Default is 0, meaning no limit. Text exceeding the given value in length will have extra characters truncated.

**Text**  
Text contents for this text box.

**Color**

**Back Color**  
Color to be used for the text box's background.

**Fore Color**  
Color to be used for the text box's foreground, the color of the text caption.

**Color Set**  
Choose from Standard Colors (standard Windows non-dithered colors) or Windows System Colors. The latter will be defined by the user's Control Panel settings and is normally the setting used so that the form changes appearance like the rest of Windows if the Control Panel settings are changed.

**Color Palette**  
Displays available colors. Click on the property to be changed to highlight it in the Properties pane, click on the desired color in the Color Palette pane and then press **Apply**.

**Edit Custom Color**  
Change the custom color presented in the Color Palette when the **Color Set** is set to Windows System Colors.
| Font Properties | Font properties that may be changes. |
| Font | Choose a font installed on this system. It's wise to choose standard Windows fonts such as MS Sans Serif that are universally available. |
| Size | Size of font, in points. |
| Effects | Bold, Italic, Underline or Strikeout. |
| Sample Text | A preview of the selected settings. |
Control - List Box

List boxes are used to display a list of items from which one or more items can be selected. If there are more items than will fit in the drop-down list, a scroll bar will be automatically added.

To create a list box that is a drop down menu style of list box, use a Combo Box control with Style set to DropdownList.

To Add a List box to a Form

1. Click on the Insert List box button in the Tools toolbar.
2. Click and drag the location and size where it is to be created in the form.
3. Continue to create any additional list boxes desired.
4. Right click onto each list box and choose Properties from the context menu.
5. Set the properties for the list box as desired.
6. Move, resize and align the list box as desired.

Properties normally both return and set that particular property. The easiest way to edit properties is through the context menu's properties dialog for that control.

Some attributes of controls, such as their size or position are properties not of the control but of the container for the control, in this case the Form itself. To change the size or position of controls we can simply click on them and resize or drag them to a new position.

Keep in mind that programs can change the properties of controls at run time. Although this documentation is written as though the properties are set at design time using the Properties dialog for this control, there are many situations in which program code will be used to change a control's properties at run time.

Note: All properties and controls in the Tools toolbar are prefaced with "wl" to indicate they are the windowless lightweight versions of the controls and properties settings. The "wl" prefix is suppressed in the documentation below to enhance legibility.

Properties

In addition to the properties listed below, a Control tab provides access to certain frequently used form properties for this control. See the Form Properties topic for information on those properties.

List box

General

MousePointer The type of mouse pointer displayed when over the list box. Choose from 16 different types plus a custom pointer

Appearance 3D or Flat. 3D is the standard Windows look. Flat results in a rectangular square of background color with a foreground color border.

MultiSelect Whether or not user can make multiple selections from the list box. NoMultiSelect means none. Simple, means a mouse click or SPACEBAR selects/deselects while arrow keys move focus. Extended adds CTRL and SHIFT click selection in the usual way.

OLEDragMode DragManual (default) or DragAutomatic. DragManual means the program handles all OLE drag/drop operations, while DragAutomatic means the list box handles all OLE drag/drop operations.

OLEDropMode Set to None (default) if the list box does not accept OLE drops and is to display the No Drop cursor if an OLE drop is attempted. Set to Manual, the list box will trigger OLE drop events, allowing programs to handle the OLE drop.
operation in code.

**Style** Standard or Checkbox. Checkbox adds little boxes that may be checked ON or OFF to select an item in the list.

**Enabled** True or False. True means the object can respond to user-generated events, false prevents it from responding. We can disable a control like a list box if we wish it to simply display information in a read-only way.

**Integral Height** True by default, so list resizes to show only complete items. False allows list size so partial items are visible.

**Sorted** True or False. True means to sort the list alphabetically.

**Columns** An integer number from 0 to n. 0 means no columns and the list box will scroll vertically. 1 to n means up to n columns and the list box will scroll horizontally through snaking columns as the columns are filled in with list items. If originally set non-zero, the number of columns may be changed at run time, but a zero setting at design time cannot be made non-zero at run time and vice versa.

**List**

**List** List items for this list box. Add an item by entering it and then pressing CTRL-ENTER to move focus to another line. Press Apply.

**ItemData** After adding a list item in the List pane and pressing Apply, a 0 value by default will be added to the ItemData pane for that item. Change this to whatever value is desired.

**Color**

**BackColor** Color to be used for the list box's background.

**ForeColor** Color to be used for the list box's foreground, the color of the text caption.

**Color Set** Choose from Standard Colors (standard Windows non-dithered colors) or Windows System Colors. The latter will be defined by the user's Control Panel settings and is normally the setting used so that the form changes appearance like the rest of Windows if the Control Panel settings are changed.

**Color Palette** Displays available colors. Click on the property to be changed to highlight it in the Properties pane, click on the desired color in the Color Palette pane and then press Apply.

**Edit Custom Color** Change the custom color presented in the Color Palette when the Color Set is set to Windows System Colors.

**Font**

**Properties** Font properties that may be changes.

**Font** Choose a font installed on this system. It's wise to choose standard Windows fonts such as MS Sans Serif that are universally available.

**Size** Size of font, in points.

**Effects** Bold, Italic, Underline or Strikeout.
Sample Text  A preview of the selected settings.
**Control - Combo Box**

Combo boxes combine the capabilities of a text box control and a list box control. Combo boxes allow the user to either enter text into the box or to choose one or more items from a list. If there are more items than will fit in the drop-down list, a scroll bar will be automatically added.

When a DropDownList Style is selected, the combo box becomes a list box that has a dropdown menu without the "combo" text box.

**To Add a Combo box to a Form**

1. Click on the Insert Combo box button in the Tools toolbar.
2. Click and drag the location and size where it is to be created in the form.
3. Continue to create any additional combo boxes desired.
4. Right click onto each combo box and choose Properties from the context menu.
5. Set the properties for the combo box as desired.
6. Move, resize and align the combo box as desired.

Properties normally both return and set that particular property. The easiest way to edit properties is through the context menu's properties dialog for that control.

Some attributes of controls, such as their size or position are properties not of the control but of the container for the control, in this case the Form itself. To change the size or position of controls we can simply click on them and resize or drag them to a new position.

Keep in mind that programs can change the properties of controls at run time. Although this documentation is written as though the properties are set at design time using the Properties dialog for this control, there are many situations in which program code will be used to change a control's properties at run time.

**Note:** All properties and controls in the Tools toolbar are prefaced with "wl" to indicate they are the windowless lightweight versions of the controls and properties settings. The "wl" prefix is suppressed in the documentation below to enhance legibility.

**Properties**

In addition to the properties listed below, a Control tab provides access to certain frequently used form properties for this control. See the Form Properties topic for information on those properties.

**Combo box**

**General**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Text</strong></td>
<td>The initial text for the combo box.</td>
</tr>
<tr>
<td><strong>MousePointer</strong></td>
<td>The type of mouse pointer displayed when over the combo box. Choose from 16 different types plus a custom pointer</td>
</tr>
<tr>
<td><strong>Appearance</strong></td>
<td>3D or Flat. 3D is the standard Windows look. Flat results in a rectangular square of background color with a foreground color border.</td>
</tr>
<tr>
<td><strong>OLEDragMode</strong></td>
<td>DragManual (default) or DragAutomatic. DragManual means the program handles all OLE drag/drop operations, while DragAutomatic means the combo box handles all OLE drag/drop operations</td>
</tr>
<tr>
<td><strong>OLEDropMode</strong></td>
<td>Set to None (default) if the combo box does not accept OLE drops and is to display the No Drop cursor if an OLE drop is attempted. Set to Manual, the combo box will trigger OLE drop events, allowing programs to handle the OLE drop operation in code.</td>
</tr>
</tbody>
</table>
Style
- DropdownCombo: classic Windows combo box.
- SimpleCombo: a text box and a list that does not drop down. Must increase the height to see the list after creating the box. DropdownList: just the dropdown list portion without the editable text box.

Enabled
True or False. True means the object can respond to user-generated events, false prevents it from responding. We can disable a control like a combo box if we wish it to simply display information in a read-only way.

Integral Height
True by default, so list resizes to show only complete items. False allows list size so partial items are visible.

Locked
True or False. True means the user can scroll and highlight the text in the box but cannot change it. The program can change the text by modifying the text box's Text property. False (default) means users can edit the text in the box.

Sorted
True or False. True means to sort the list alphabetically.

List
- List: List items for this combo box. Add an item by entering it and then pressing CTRL-ENTER to move focus to another line. Press Apply.
- ItemData: After adding a list item in the List pane and pressing Apply, a 0 value by default will be added to the ItemData pane for that item. Change this to whatever value is desired.

Color
- Back Color: Color to be used for the combo box's background.
- Fore Color: Color to be used for the combo box's foreground, the color of the text caption.
- Color Set: Choose from Standard Colors (standard Windows non-dithered colors) or Windows System Colors. The latter will be defined by the user's Control Panel settings and is normally the setting used so that the form changes appearance like the rest of Windows if the Control Panel settings are changed.
- Color Palette: Displays available colors. Click on the property to be changed to highlight it in the Properties pane, click on the desired color in the Color Palette pane and then press Apply.
- Edit Custom Color: Change the custom color presented in the Color Palette when the Color Set is set to Windows System Colors.

Font
- Properties: Font properties that may be changes.
- Font: Choose a font installed on this system. It's wise to choose standard Windows fonts such as MS Sans Serif that are universally available.
- Size: Size of font, in points.
- Effects: Bold, Italic, Underline or Strikeout.
Sample Text  A preview of the selected settings.
Control - Horiz / Vert Scroll Bars

Scroll bars allow navigation through lists of many items or through large amounts of information. They are also used to scroll through pictures or otherwise move position or to change values in an apparently analog way.

To Add a Scroll bar to a Form

1. Click on the Insert Horizontal Scroll bar or Insert Vertical Scroll bar button in the Tools toolbar.
2. Click and drag the location and size where it is to be created in the form.
3. Continue to create any additional scroll bars desired.
4. Right click onto each scroll bar and choose Properties from the context menu.
5. Set the properties for the scroll bar as desired.
6. Move, resize and align the scroll bar as desired.

Properties normally both return and set that particular property. The easiest way to edit properties is through the context menu’s properties dialog for that control.

Some attributes of controls, such as their size or position are properties not of the control but of the container for the control, in this case the Form itself. To change the size or position of controls we can simply click on them and resize or drag them to a new position.

Keep in mind that programs can change the properties of controls at run time. Although this documentation is written as though the properties are set at design time using the Properties dialog for this control, there are many situations in which program code will be used to change a control's properties at run time.

Note: All properties and controls in the Tools toolbar are prefaced with "wl" to indicate they are the windowless lightweight versions of the controls and properties settings. The "wl" prefix is suppressed in the documentation below to enhance legibility.

Properties

In addition to the properties listed below, a Control tab provides access to certain frequently used form properties for this control. See the Form Properties topic for information on those properties.

Scroll bar

General

MousePointer The type of mouse pointer displayed when over the scroll bar. Choose from 16 different types plus a custom pointer

Min Minimum value returned by the scroll bar, from zero (default).

Max Maximum value returned by the scroll bar, to 32767 (default).

SmallChange The amount of change in scroll bar value when user clicks on scroll arrow.

LargeChange The amount of change in scroll bar value when user clicks in region between the scroll box and scroll arrow.

Value Current position of the scroll bar, between Min and Max, inclusive.

Enabled True or False. True means the object can respond to user-generated events, false prevents it from responding. We can disable a control like a scroll bar if we wish it to simply display information in a read-only way.
Advanced Controls

Control - Tools (Advanced) Toolbar

The Tools (Advanced) toolbar is used with Forms to create frequently used Windows standard user interface controls that are technically more advanced or less frequently used than those in the Tools toolbar. The toolbar is enabled when a Form is the active window. Forms are used to provide user interfaces to scripts, to view tables and for other functions within Manifold.

Click on a control in the Tools (Advanced) toolbar and then click and drag in the Form window to insert the control in the form in the position and size indicated by the mouse click and drag. Forms are then customized using the properties for each control.

See the individual controls topics for details on each control. The individual control topics are the briefest summary of the control properties that appear in the Manifold dialogs and should be supplemented by reference to any one of the numerous commercial texts on use of standard Windows controls from VB.

Properties listed in the individual controls topics are only those properties that appear at design time in the Properties dialog. Advanced controls typically have other properties that are set or read at run time. Properties in advanced controls that are typical to Windows Common Controls are prefaced with "cc", suppressed in the following topics for greater legibility. Prefixes are obvious from inspection of the options in the Properties dialogs.

Tools (Advanced) Controls

<table>
<thead>
<tr>
<th>Insert Animation</th>
<th>Plays silent .avi files to create dialog effects like the sheet of paper that flies between folders in the Windows copy progress dialog. Can also play silent .avi files dropped onto the control. Manifold provides 19 cool Windows dialog .avi files to get started.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Insert Image List</td>
<td>Contains a collection of images that can be used by other Windows Common Controls such as ListView, TreeView, TabStrip and Toolbar controls, as well as with other controls with a Picture property. Saves development time by maintaining all images in a standard, consistent catalog of images.</td>
</tr>
<tr>
<td>Insert Image Combo Box</td>
<td>Similar to standard combo but with ability to include pictures with each item in the list portion of the combo, with special facilities for list management.</td>
</tr>
<tr>
<td>Insert Date / Time Picker</td>
<td>Displays date and/or time info and provides an interface for modifying date and time. Dropdown menu provides a MonthView calendar.</td>
</tr>
<tr>
<td>Insert Month View</td>
<td>An easy way to view and set date information using a monthly calendar. Can select one or multiple dates or show up to 12 months at a time.</td>
</tr>
<tr>
<td>Insert List View</td>
<td>Displays data as ListItem objects with an optional icon. Four different views provide data as icons, small icons, a list or a report. A sophisticated control.</td>
</tr>
<tr>
<td>Insert Progress Bar</td>
<td>Create a progress bar display that can be used to show the progress of a process.</td>
</tr>
</tbody>
</table>
| Insert Masked Text Box | Used to prompt users for input using a fixed format specified by a mask pattern, for example, a telephone number in a specified format. If a
mask pattern is not used, this control behaves about the same as a standard text box.

**Insert Rich Text Box** - Allows entry and editing of text using advanced formatting features, such as paragraph formatting with left and right indents, hanging indents, bold and italic font and so on. It also supports object embedding using the OLEObjects collection. Write your own word processor or create elaborate forms for users.

**Insert Slider** - Insert a slider bar control that can be moved left and right.

**Insert Tree View** - Insert a hierarchical tree display. Designed to display data such as organization trees, entries in an index, files and directories on disk, etc.

**Insert UpDown** - Insert an increment/decrement control, also called **spin buttons**.

**Insert Tab Strip** - Insert a tabbed control where clicking on each tab brings it to the fore.

**Insert Picture Clip** - Displays graphics from bitmap, icon, metafile, JPEG or GIF files.

**Insert Multimedia Control** - Manages Media Control Interface (MCI) devices like sound boards, MIDI sequencers, CD-ROM drives, audio players, videodisc players, etc. Add voice note recording to your dialog, or play spoken announcements.

**Insert Status Bar** - Insert a status bar like those at the bottom of many program windows. Complete with a nearly infinite set of properties for the intrepid programmer.

**Insert Tool Bar** - Insert a toolbar at the top of the form that hosts buttons or other controls. Yet another control with a vast array of properties.

**Insert Cool Bar** - Requires installation of Internet Explorer 3.0 or greater. Provides a modern, "railbar" look and the ability to create (get this!) **user configurable** toolbars like those in IE.

**Insert Common Dialog Control** - Provides a standard set of Windows dialog boxes for opening and saving files and selecting colors and fonts. Can also display Help.

**Insert System Info Control** - Detects system events such as desktop resizing, resolution changes, time changes. Also provides operating system platform and version information and changes in AC/battery power status and Plug/Play hardware configuration. Used when writing applications for portable devices (GIS in the field) and for developing info for your tech support team.

**Insert ActiveX Control** - Insert any ActiveX control available on this system.

**Other Controls**

Frequently used controls appear in the Tools toolbar for Form windows. Read the introductory topic before using those controls since they are implemented using a special, windowless technology that requires attention from the programmer.

**Form Properties for Controls**
Manifold adds an extra tab, the Control tab, to the usual tabs that appear in the design time properties dialog for controls. This tab provides access to the properties for this control maintained by the form. This extra tab will appear at design time for all controls. Although strictly speaking these are properties of the form that is the container for the control and not of the control, they are easiest to specify at design time if they appear within each control's properties dialog.

Control

<table>
<thead>
<tr>
<th>Name</th>
<th>The name used by the form to identify this control.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tag</td>
<td>A string available for free use by the programmer. Often used for comments.</td>
</tr>
<tr>
<td>Left</td>
<td>Left margin position of the control.</td>
</tr>
<tr>
<td>Top</td>
<td>Top margin position of the control.</td>
</tr>
<tr>
<td>Width</td>
<td>Width of the control.</td>
</tr>
<tr>
<td>Height</td>
<td>Height of the control.</td>
</tr>
<tr>
<td>Visible</td>
<td>Make this control visible or invisible.</td>
</tr>
</tbody>
</table>
Control - Animation

Plays silent .avi files to create dialog effects like the sheet of paper that flies between folders in the Windows copy progress dialog. Can also play silent .avi files dropped onto the control. Manifold provides 19 cool Windows dialog .avi files to get started.

Properties

In addition to the properties listed below, a Control tab provides access to certain frequently used form properties for this control. See the Form Properties topic for information on those properties.

Animation

General

**OLEDropMode**
Set to None (default) if the animation does not accept OLE drops and is to display the No Drop cursor if an OLE drop is attempted. Set to Manual, the animation will trigger OLE drop events, allowing programs to handle the OLE drop operation in code.

**AutoPlay**
True - the .avi file plays automatically in a continuous loop once it is loaded into the control. False - Does not play until the Play method is used.

**Center**
True (Default) - Center the .avi within the control. False - position the .avi at 0,0.

**BackStyle**
Transparent (Default) - background color of the control is shown. Opaque - background color from the .avi fills the control and covers the control's background color.

Color

**Back Color**
Color to be used for the animation box background.

**Color Set**
Choose from Standard Colors (standard Windows non-dithered colors) or Windows System Colors. The latter will be defined by the user's Control Panel settings and is normally the setting used so that the form changes appearance like the rest of Windows if the Control Panel settings are changed.

**Color Palette**
Displays available colors. Click on the property to be changed to highlight it in the Properties pane, click on the desired color in the Color Palette pane and then press Apply.

**Edit Custom Color**
Change the custom color presented in the Color Palette when the Color Set is set to Windows System Colors.
Control - Chart

Displays a chart.
Control - Image List

Contains a collection of images that can be used by other Windows Common Controls such as ListView, TreeView, TabStrip and Toolbar controls, as well as with other controls with a Picture property. Saves development time by maintaining all images in a standard, consistent catalog of images.

ImageList can use bitmap (.bmp), cursor (.cur), icon (.ico), JPEG (.jpg) or GIF (.gif) files for images. We can add/remove images at design time or at run time using the standard collection object properties (Key and Index) and the standard methods (Add, Remove, Clear, etc).

Use ImageList to store images representing open/closed folders, documents, etc for use in a TreeView. These can be dynamically assigned to the TreeView nodes as it expands or collapses. ImageList is a good place to save frequently used iconic images such as file open/close/print images for toolbar buttons or MousePointer icons and DragIcons.

To Add Images at Design Time

1. Click Insert Picture in the control's Images property tab.
2. Use the resulting Select Picture dialog to find the desired image file and open it (multiple files may be opened).
3. In the Key box assign a unique string for each image.
4. If desired, optionally assign a Tag string for each image. Tag strings need not be unique.

Properties

In addition to the properties listed below, a Control tab provides access to certain frequently used form properties for this control. See the Form Properties topic for information on those properties.

Command Button

General

16 x 16, 32 x 32, 48 x 48, Custom

Image sizes in this control. Must be specified before any images are added. The size of the first image is what is used for all images when feeding images to a Windows Common Control bound to this one.

Height, Width

Size of image to use when Custom size is enabled.

UseMaskColor

If checked, enables use of mask color in a. A mask color is the color that is to be made transparent.

Images

Index

Integer that uniquely identifies an image in the ImageList collection. Numbered in order of creation from the first image, which is always 1.

Key

A unique string that identifies each image.

Tag

Stores a string available for the programmer to use as desired. Often used to store an identification string of some sort.

Images

Previews of the images in the ListImage collection. Add with InsertPicture and remove with RemovePicture.

Image Count

…Gosh… What could this number be?
<table>
<thead>
<tr>
<th><strong>Back Color</strong></th>
<th>Color to be used for image's background, if any.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Mask Color</strong></td>
<td>Color to be treated as transparent in an image used when UseMaskColor is checked.</td>
</tr>
<tr>
<td><strong>Color Set</strong></td>
<td>Choose from Standard Colors (standard Windows non-dithered colors) or Windows System Colors. The latter will be defined by the user's Control Panel settings and is normally the setting used so that the form changes appearance like the rest of Windows if the Control Panel settings are changed.</td>
</tr>
<tr>
<td><strong>Color Palette</strong></td>
<td>Displays available colors. Click on the property to be changed to highlight it in the Properties pane, click on the desired color in the Color Palette pane and then press Apply.</td>
</tr>
<tr>
<td><strong>Edit Custom Color</strong></td>
<td>Change the custom color presented in the Color Palette when the Color Set is set to Windows System Colors.</td>
</tr>
</tbody>
</table>
Control - Image Combo Box

Similar to standard combo but with ability to include pictures with each item in the list portion of the combo, with special facilities for list management.

Combo boxes combine the capabilities of a text box control and a list box control. Combo boxes allow the user to either enter text into the box or to choose one or more items from a list. If there are more items than will fit in the drop-down list, a scroll bar will be automatically added. Image Combo boxes allow indentation of the items within them.

Image Combo boxes must be used in conjunction with an associated Image List control, which contains the images to be used for the Image Combo box’s items. Unlike regular combo boxes, which may be populated at design time using their List tab, an Image Combo box is populated with items programmatically at run time. Only the choice of which Image List is associated with the Image Combo box is made at design time.

To Add an Image Combo box to a Form

1. Click on the Insert Image Combo box button in the Tools toolbar.
2. Click and drag the location and size where it is to be created in the form.
3. Continue to create any additional combo boxes desired.
4. Right click onto each combo box and choose Properties from the context menu.
5. Set the properties for the combo box as desired.
6. Move, resize and align the combo box as desired.
7. Create and populate with images the Image List associated with this Image Combo.
8. Set the ImageList property to the associated Image List.
9. At run time, populate the Image Combo with items.

Properties

In addition to the properties listed below, a Control tab provides access to certain frequently used form properties for this control. See the Form Properties topic for information on those properties.

Combo box

General

MousePointer The type of mouse pointer displayed when over the combo box. Choose from 16 different types plus a custom pointer.

ImageList Associated ImageList control.

OLEDragMode DragManual (default) or DragAutomatic. DragManual means the program handles all OLE drag/drop operations, while DragAutomatic means the combo box handles all OLE drag/drop operations.

OLERefMode Set to None (default) if the combo box does not accept OLE drops and is to display the No Drop cursor if an OLE drop is attempted. Set to Manual, the combo box will trigger OLE drop events, allowing programs to handle the OLE drop operation in code.

Indentation Starting indentation for items in the drop down list.

Text Initial text string for the box.

Enabled True or False. True means the object can respond to user-generated events, false prevents it from responding. We can disable a control like a combo box if we wish it to simply display information in a read-only way.

Locked True or False. True means the user can scroll and highlight the text in the box but cannot change it.
program can change the text by modifying the text box's **Text** property. False (default) means users can edit the text in the box.

### Color

**Back Color**  
Color to be used for the combo box's background.

**Fore Color**  
Color to be used for the combo box's foreground, the color of the text caption.

**Color Set**  
Choose from Standard Colors (standard Windows nondithered colors) or Windows System Colors. The latter will be defined by the user's Control Panel settings and is normally the setting used so that the form changes appearance like the rest of Windows if the Control Panel settings are changed.

**Color Palette**  
Displays available colors. Click on the property to be changed to highlight it in the **Properties** pane, click on the desired color in the **Color Palette** pane and then press **Apply**.

**Edit Custom Color**  
Change the custom color presented in the **Color Palette** when the **Color Set** is set to Windows System Colors.

### Font

**Properties**  
Font properties that may be changes.

**Font**  
Choose a font installed on this system. It's wise to choose standard Windows fonts such as **MS Sans Serif** that are universally available.

**Size**  
Size of font, in points.

**Effects**  
Bold, Italic, Underline or Strikeout.

**Sample Text**  
A preview of the selected settings.

### Picture

**MouseIcon**  
Custom icon to use when the **MousePointer** property is set to 99. Like Visual Basic, will not load animated cursor (.ani) files.
Control - Date / Time Picker

Displays date and/or time info and provides an interface for modifying date and time. Dropdown menu provides a MonthView calendar.

The UpDown property sets whether this control operates as a masked edit box for dates and times, or if it operates with a dropdown calendar. Setting UpDown true adds an UpDown control to the box and suppresses the calendar. Using the UpDown button will tab the focus between each part of the date and time mask for entry of that part of the date/time.

Properties

In addition to the properties listed below, a Control tab provides access to certain frequently used form properties for this control. See the Form Properties topic for information on those properties.

DateTimePicker

General

Value  Date expression that is the default property of the control. Set a default date, if desired.

MinDate  Date expression setting lower boundary of the calendar.

MaxDate  Date expression setting upper boundary of the calendar.

Format  LongDate, ShortDate, Time or Custom.

MousePointer  The type of mouse pointer displayed when over the DateTimePicker. Choose from 16 different types plus a custom pointer

OLEDropMode  Set to None (default) if the DateTimePicker does not accept OLE drops and is to display the No Drop cursor if an OLE drop is attempted. Set to Manual, the DateTimePicker will trigger OLE drop events, allowing programs to handle the OLE drop operation in code.

CustomFormat  string setting forth custom format when Format is set to custom.

Enabled  True or False. True means the object can respond to user-generated events, false prevents it from responding.

CheckBox  False (Default) - Control always returns a date. True - A checkbox appears. If the user does not check this box, no date is returned. Handy for forms where requiring a date ("Due date?") is an option.

UpDown  True - UpDown control appears to change the date instead of a drop down calendar. False (Default) - Calendar appears. When True, switches mode to become a masked edit control.

Time  Sets Hour, Minute and Second properties.

Font

Properties  Font properties that may be changes.

Font  Choose a font installed on this system. It's wise to choose standard Windows fonts such as MS Sans Serif that are universally available.

Size  Size of font, in points. The size of the font sets the size of the drop down portion of this control.
Effects  Bold, Italic, Underline or Strikeout.

Sample Text  A preview of the selected settings.

**Color**

**CalendarBackColor**  Color to be used for the DateTimePicker's background.

**CalendarForeColor**  Color to be used for the DateTimePicker's foreground, the color of the text caption.

**CalendarTitleBack Color**  Customize calendar title background color.

**CalendarTitleFore Color**  Customize calendar title foreground color.

**CalendarTrailingForeColor**  Color of dates that preceded and follow the displayed month or months.

**Color Set**  Choose from Standard Colors (standard Windows non-dithered colors) or Windows System Colors. The latter will be defined by the user's Control Panel settings and is normally the setting used so that the form changes appearance like the rest of Windows if the Control Panel settings are changed.

**Color Palette**  Displays available colors. Click on the property to be changed to highlight it in the Properties pane, click on the desired color in the Color Palette pane and then press **Apply**.

**Edit Custom Color**  Change the custom color presented in the Color Palette when the **Color Set** is set to Windows System Colors.

**Picture**

**MouseIcon**  Custom icon to use when the MousePointer property is set to 99. Like Visual Basic, will not load animated cursor (.ani) files.

**Custom Date and Time Formats**

The **CustomFormat** property of the **DateTimePicker** allows specifying fully custom formats for dates and times that can intermingle regular text and the following special format characters within the mast string.

- **d**  One- or two-digit day.
- **dd**  Two-digit days with single digits prefixed with 0.
- **ddd**  Three-character abbreviated weekday.
- **dddd**  Full weekday name.
- **h**  One- or two-digit hour in 12-hour format.
- **hh**  Two-digit hour in 12-hour format with single digit hours prefixed with 0.
- **H**  One- or two digit hour in 24-hour format.
- **HH**  Two-digit hour in 24-hour format with single digit hours prefixed with 0.
- **m**  One- or two-digit minute.
- **mm**  Two-digit minutes with single digits prefixed with 0.
Manifold® System Release 8.00 User Manual

M  One- or two-digit month number.

MM  Two-digit month numbers with single digits prefixed with 0.

MMM  Three-character month name abbreviation.

MMMM  Full month name.

s  One- or two-digit second.

ss  Two-digit seconds with single digits prefixed with 0.

t  One-letter AM/PM abbreviation: either "A" or "P"

tt  Either "AM" or "PM"

X  A callback field. The control fills in all other fields and then asks the owner how to fill in the portion with "X" characters. Use different numbers of X's (like, XX or XXXX for different call back fields in the same mask string).

y  The last digit of the year.

yy  The last two digits of the year.

yyy  The full year.

Example:

"The date is 'ddddMMMMdd, yyy ' and the time is 'h:mtt"

Will result in dates such as:

"The date is Friday February 04, 2000 and the time is 3:34 PM"

Callbacks are used with format strings like “dddd XX ” to see what should be put in the XX based on what is in the dddd, for example to substitute the XX with a non-English day name in addition to the English name. If the dddd is "Wednesday" you could write the appropriate non-English name for the target audience. Consult a good VB reference text for use of callbacks with this control.

Notes

- More than one month may be displayed at a time by setting MonthRows and MonthColumns properties at run time.
- Determine which day of the week appears as the first day at run time by changing the value of the StartOfWeek property.
- Individual days can have their day number boldfaced at run time to emphasize holidays, etc.
Control - Month View

An easy way to view and set date information using monthly calendars. Can select one or multiple dates or show up to 12 months at a time.

Properties

In addition to the properties listed below, a Control tab provides access to certain frequently used form properties for this control. See the Form Properties topic for information on those properties.

MonthView

General

**Value**
- Date expression that is the default property of the control. Set a default date, if desired.

**MinDate**
- Date expression setting lower boundary of the calendar.

**MaxDate**
- Date expression setting upper boundary of the calendar.

**StartOfWeek**
- Day of the week to use to start each week.

**MousePointer**
- The type of mouse pointer displayed when over the MonthView. Choose from 16 different types plus a custom pointer.

**OLEDropMode**
- Set to None (default) if the MonthView does not accept OLE drops and is to display the No Drop cursor if an OLE drop is attempted. Set to Manual, the MonthView will trigger OLE drop events, allowing programs to handle the OLE drop operation in code.

**BorderStyle**
- FixedSingle (default) or None. None may be used with Flat Appearance to achieve an uncaptioned rectangle of background color.

**Appearance**
- 3D or Flat. 3D is the standard Windows look. Flat results in a rectangular square of background color with a foreground color border.

**ShowWeekNumbers**
- Show week number (week position in year) along left column border.

**MultiSelect**
- Whether or not user can make multiple selections from the MonthView.

**Enabled**
- True or False. True means the object can respond to user-generated events, false prevents it from responding.

**ShowToday**
- True to show “Today is” date line at bottom of calendar.

**MonthRows**
- How many rows to use to show multiple months.

**MonthColumns**
- How many columns to use to show multiple months.

**ScrollRate**
- The number of months that are scrolled when user clicks a month scroll button. Allows scrolling of more than one month at a time, so if four months are displayed an even four months at a time can be scrolled.

**MaxSelCount**
- Valid when MultiSelect is True. Sets the maximum number of contiguous days that can be selected at once.

Font

Properties
- Font properties that may be changes.
Font  Choose a font installed on this system. It's wise to choose standard Windows fonts such as **MS Sans Serif** that are universally available.

Size  Size of font, in points. The size of the font sets the size of the drop down portion of this control.

Effects  Bold, Italic, Underline or Strikeout.

Sample Text  A preview of the selected settings.

**Color**

BackColor  Color to be used for the MonthView's background.

ForeColor  Color to be used for the MonthView's foreground, the color of the text caption.

MonthBackColor  Background color for months.

TitleBackColor  Customize calendar title background color.

TitleForeColor  Customize calendar title foreground color.

TrailingForeColor  Color of dates that preceded and follow the displayed month or months.

Color Set  Choose from Standard Colors (standard Windows non-dithered colors) or Windows System Colors. The latter will be defined by the user's Control Panel settings and is normally the setting used so that the form changes appearance like the rest of Windows if the Control Panel settings are changed.

Color Palette  Displays available colors. Click on the property to be changed to highlight it in the Properties pane, click on the desired color in the Color Palette pane and then press Apply.

Edit Custom Color  Change the custom color presented in the Color Palette when the Color Set is set to Windows System Colors.

**Picture**

MouseIcon  Custom icon to use when the MousePointer property is set to 99. Like Visual Basic, will not load animated cursor (.ani) files.

**Custom Date and Time Formats**

The CustomFormat property of the MonthView allows specifying fully custom formats for dates and times that can intermingle regular text and the following special format characters within the mast string.

- **d**  One- or two-digit day.
- **dd**  Two-digit days with single digits prefixed with 0.
- **ddd**  Three-character abbreviated weekday.
- **dddd**  Full weekday name.
- **h**  One- or two-digit hour in 12-hour format.
- **hh**  Two-digit hour in 12-hour format with single digit hours prefixed with 0.
H  One- or two digit hour in 24-hour format.
HH  Two-digit hour in 24-hour format with single digit hours prefixed with 0.
m  One- or two-digit minute.
mm  Two-digit minutes with single digits prefixed with 0.
M  One- or two-digit month number.
MM  Two-digit month numbers with single digits prefixed with 0.
MMM  Three-character month name abbreviation.
MMMM  Full month name.
s  One- or two-digit second.
ss  Two-digit seconds with single digits prefixed with 0.
t  One-letter AM/PM abbreviation: either "A" or "P"
tt  Either "AM" or "PM"
X  A callback field. The control fills in all other fields and then asks the owner how to fill in the portion with "X" characters. Use different numbers of X's (like, XX or XXXX for different call back fields in the same mask string).
y  The last digit of the year.
yy  The last two digits of the year.
yyy  The full year.

**Example:**

"'The date is 'ddddMMMMdd, yyy ' and the time is 'h:mtt"

Will result in dates such as:

"The date is Friday February 04, 2000 and the time is 3:34 PM"

Callbacks are used with format strings like "dddd XX " to see what should be put in the XX based on what is in the dddd, for example to substitute the XX with a non-English day name in addition to the English name. If the dddd is "Wednesday" you could write the appropriate non-English name for the target audience. Consult a good VB reference text for use of callbacks with this control.

**Note:** Individual days can have their day number boldfaced at run time to emphasize holidays, etc.
Control - List View

Displays data as ListItem objects with an optional icon. Four different views provide data as icons, small icons, a list or a report. Can show data in columns with or without column heads. A very sophisticated control.

Note: Non-Common Control properties are prefaced with "lvw", suppressed in the following for greater legibility.

Properties

In addition to the properties listed below, a Control tab provides access to certain frequently used form properties for this control. See the Form Properties topic for information on those properties.

List View

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MousePointer</td>
<td>The type of mouse pointer displayed when over the list view. Choose from 16 different types plus a custom pointer</td>
</tr>
<tr>
<td>View</td>
<td>Style of display for each item: Icon - Full size icon and text label. SmallIcon - small icon and text label arranged horizontally. List - small icon and text label arranged vertically in columns. Report - small icons and text labels arranged in columns with additional info columns as desired.</td>
</tr>
<tr>
<td>Arrange</td>
<td>None, AutoLeft, AutoTop: Left and top align items along the left side or top of the control.</td>
</tr>
<tr>
<td>LabelEdit</td>
<td>Automatic - BeforeLabelEdit event is generated when the user clicks the label of a selected node. Manual - BeforeLabelEdit event generated only when the StartLabelEdit method is invoked.</td>
</tr>
<tr>
<td>BorderStyle</td>
<td>FixedSingle (default) or None. None may be used with Flat Appearance to achieve an uncaptioned rectangle of background color.</td>
</tr>
<tr>
<td>Appearance</td>
<td>3D or Flat. 3D is the standard Windows look. Flat results in a rectangular square of background color with a foreground color border.</td>
</tr>
<tr>
<td>OLEDragMode</td>
<td>DragManual (default) or DragAutomatic. DragManual means the program handles all OLE drag/drop operations, while DragAutomatic means the list view handles all OLE drag/drop operations</td>
</tr>
<tr>
<td>OLEDropMode</td>
<td>Set to None (default) if the list view does not accept OLE drops and is to display the No Drop cursor if an OLE drop is attempted. Set to Manual, the list view will trigger OLE drop events, allowing programs to handle the OLE drop operation in code.</td>
</tr>
<tr>
<td>HideColumnHeaders</td>
<td>False (default) - Column headers are visible. True - column headers are hidden.</td>
</tr>
<tr>
<td>HideSelection</td>
<td>True or False. True means that highlighted text will not appear highlighted when the list view loses the focus. False means that highlighted text stays highlighted even when another form or dialog has the focus.</td>
</tr>
<tr>
<td>LabelWrap</td>
<td>True (Default) - labels wrap. False - labels don't wrap.</td>
</tr>
<tr>
<td>MultiSelect</td>
<td>Whether or not user can make multiple selections from the list view. NoMultiSelect means none. Simple, means a mouse click or SPACEBAR selects/deselects while arrow keys move focus. Extended adds CTRL and SHIFT click selection in the usual way.</td>
</tr>
<tr>
<td>Enabled</td>
<td>True or False. True means the object can respond to user-</td>
</tr>
</tbody>
</table>
generated events, false prevents it from responding. We can disable a control like a list view if we wish it to simply display information in a read-only way.

**AllowColumnReorder**
- False (Default) - User cannot reorder columns.
- True - Users can reorder columns.

**Checkboxes**
- False (Default) - Checkboxes do not appear.
- True - Checkboxes appear.

**FlatScrollbar**
- False (Default) - Standard scrollbars appear.
- True - FlatScrollBar style scrollbars appear.

**FullScreenSelect**
- Specifies if the entire row is selected. Valid only when View is set to Report.

**Gridlines**
- Specifies if gridlines appear when View is set to Report.
- False (Default) - No gridlines.
- True - Display gridlines.

**HotTracking**
- Whether or not mouse-sensitive highlighting is enabled.
- True - Header captions are highlighted as the mouse passes over them.
- False - HeaderItem objects do not respond to mouse movement unless the mouse button is clicked.

**HoverSelection**
- Whether or not a ListItem object is select on a mouse hover.
- False (Default) - No selection when the mouse hovers over an item.
- True - Item is selected on mouse hover.

**Image Lists**
- **Normal** ImageList to use for images of standard sized icons.
- **Small** ImageList to use for small icons.
- **Column Header** ImageList to use for images in column headers.

**Sorting**
- **Sorted** True - Items are sorted according to the SortOrder property.
- False - Items are not sorted.

**SortKey** A positive integer or zero.
- 0 - Sort using the ListItem object’s Text property.
- >=1 - Sort using subitem whose collection index is specified here.

**SortOrder** Ascending - From beginning of alphabet, earliest date or lowest number.
- Descending - The other way.

**Column Headers**
- **Index** Uniquely identifies the column. If changing columns dynamically, use the Key since the index might change. Begins with 1.

**Insert/Remove Column** Add or remove columns at design time.

**Text** Caption for this column.

**Alignment**
- Left - Text in columns is aligned left.
- Right - Text aligned right.
- Center - Text is centered.

**Width** Width of this column.

**Key** A unique string that identifies each column.
Tag
Stores a string available for the programmer to use as desired. Often used to store an identification or description string of some sort.

Icon Index
Which image to use in the Column Header ImageList to use for this column. Begins with 0.

Color

Back Color
Color to be used for the list view's background.

Fore Color
Color to be used for the list view's foreground, the color of the text caption.

Color Set
Choose from Standard Colors (standard Windows non-dithered colors) or Windows System Colors. The latter will be defined by the user's Control Panel settings and is normally the setting used so that the form changes appearance like the rest of Windows if the Control Panel settings are changed.

Color Palette
Displays available colors. Click on the property to be changed to highlight it in the Properties pane, click on the desired color in the Color Palette pane and then press Apply.

Edit Custom Color
Change the custom color presented in the Color Palette when the Color Set is set to Windows System Colors.

Font

Properties
Font properties that may be changes.

Font
Choose a font installed on this system. It's wise to choose standard Windows fonts such as MS Sans Serif that are universally available.

Size
Size of font, in points.

Effects
Bold, Italic, Underline or Strikeout.

Sample Text
A preview of the selected settings.

Picture

MouselIcon
Custom icon to use when the MousePointer property is set to 99. Like Visual Basic, will not load animated cursor (.ani) files.

Picture
Graphic to show on an enabled control.

Note: graphics are loaded at design time and saved within the project. When browsing to find the graphic to be used, .bmp, .ico, .wmf, .gif or .jpeg files may be used.
Control - Progress Bar

Create a progress bar display that can be used to show the progress of a process.

Properties

In addition to the properties listed below, a Control tab provides access to certain frequently used form properties for this control. See the Form Properties topic for information on those properties.

Progress bar

General

Min Minimum value returned by the progress bar, from zero (default).

Max Maximum value returned by the progress bar, typically to 100 (default).

MousePointer The type of mouse pointer displayed when over the progress bar. Choose from 16 different types plus a custom pointer

BorderStyle FixedSingle (default) or None. None may be used with Flat Appearance to achieve an uncaptioned rectangle of background color.

Appearance 3D or Flat. 3D is the standard Windows look. Flat results in a rectangular square of background color with a foreground color border.

OLEDropMode Set to None (default) if the progress bar does not accept OLE drops and is to display the No Drop cursor if an OLE drop is attempted. Set to Manual, the progress bar will trigger OLE drop events, allowing programs to handle the OLE drop operation in code.

Orientation Horizontal (Default) or Vertical progress bar.

Scrolling Standard - Progress bar grows in chunks in the usual way. Make the progress bar at least 12 times wider than it is high for a reasonable effect. Smooth - A solid bar that grows smoothly.

Enabled True or False. True means the object can respond to user-generated events, false prevents it from responding. We can disable a control like a progress bar if we wish it to simply display information in a read-only way.

Picture

MouseIcon Custom icon to use when the MousePointer property is set to 99. Like Visual Basic, will not load animated cursor (.ani) files.

Tips

Set the Visible property to False to hide the progress bar until the process on which it reports agonizingly slow progress begins, and then set the Visible property to True to show the progress bar. When it is done, set the Visible property to False again.

Big fun: the progress bar displays whatever the Value is so you can make it run backwards by decrementing Value at regular intervals. Use this capability with restraint, as it is not good karma to play with people's minds by having progress bars go back and forth. Backward running progress bars are sometimes used to show de-installation or file removal progress.
Control - Masked Text Box

Used to prompt users for input using a fixed format specified by a mask pattern, for example, a telephone number in a specified format. If a mask pattern is not used, this control behaves about the same as a standard text box.

Note: Non-Common Control properties are prefaced with "msk", suppressed in the following for greater legibility.

Properties

In addition to the properties listed below, a Control tab provides access to certain frequently used form properties for this control. See the Form Properties topic for information on those properties.

Masked Edit box

General

AutoTab Whether or not the next control in the tab order receives the focus as soon as the Text property of the Masked Edit box is filled with valid data. False (Default) - AutoTab is not on. A ValidationError event occurs if more characters are entered than allowed by the input mask. True - AutoTab is on and the focus moves immediately to the next control, which then receives any extra characters entered.

PromptInclude Whether the Text property contains the prompt character. True (Default) - Text contains prompt characters, if any. False - Text property does not contain prompt characters.

HideSelection True or False. True means that highlighted text will not appear highlighted when the masked edit box loses the focus. False means that highlighted text stays highlighted even when another form or dialog has the focus.

Enabled True or False. True means the object can respond to user-generated events, false prevents it from responding.

AllowPrompt Determines if prompt character is a valid input character. False (Default) - Prompt character is not a valid input character. True - Prompt character is a valid input.

Mask Enter literal characters to be used as a mask, or choose from presets in the Format box.

Format Choose from a set of predefined formats to specify how the data in the MaskedEdit box displays and prints. Can be different than the input mask (weird!).

MousePointer The type of mouse pointer displayed when over the masked edit box. Choose from 16 different types plus a custom pointer.

BorderStyle FixedSingle (default) or None. None may be used with Flat Appearance to achieve an uncaptioned rectangle of background color.

ClipMode Whether or not literal characters are included when doing a cut or copy command. True will limit copy operations to only data entered by the user, without literals.

OLEDragMode DragManual (default) or DragAutomatic. DragManual means the program handles all OLE drag/drop operations, while DragAutomatic means the masked edit box handles all OLE drag/drop operations.

OLEDropMode Set to None (default) if the masked edit box does not accept OLE drops and is to display the No Drop cursor if an OLE drop is attempted. Set to Manual, the masked edit box will trigger OLE drop events, allowing programs...
to handle the OLE drop operation in code.

**MaxLength**  From 1 to 64. A mask field can have up to 64 characters.

**PromptChar**  The character to use for prompting users, almost always an underscore character.

### Color

**Back Color**  Color to be used for the masked edit box's background.

**Fore Color**  Color to be used for the masked edit box's foreground, the color of the text caption.

**Color Set**  Choose from Standard Colors (standard Windows non-dithered colors) or Windows System Colors. The latter will be defined by the user's Control Panel settings and is normally the setting used so that the form changes appearance like the rest of Windows if the Control Panel settings are changed.

**Color Palette**  Displays available colors. Click on the property to be changed to highlight it in the Properties pane, click on the desired color in the Color Palette pane and then press **Apply**.

**Edit Custom Color**  Change the custom color presented in the Color Palette when the Color Set is set to Windows System Colors.

### Font

**Properties**  Font properties that may be changes.

**Font**  Choose a font installed on this system. It's wise to choose standard Windows fonts such as **MS Sans Serif** that are universally available.

**Size**  Size of font, in points.

**Effects**  Bold, Italic, Underline or Strikeout.

**Sample Text**  A preview of the selected settings.

### Picture

**MouseIcon**  Custom icon to use when the MousePointer property is set to 99. Like Visual Basic, will not load animated cursor (.ani) files.

### Masks

Each character position in the input mask maps to a placeholder of a specified type or to a literal character. For example, a (###) mask has two literal characters () surrounding three digit placeholder characters, the #. As the user enters data the insertion point will automatically skip over literals to the next placeholder position indicated by the prompt character.

**Mask Characters**

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#  Digit placeholder.</td>
</tr>
<tr>
<td>.  Decimal placeholder. Actual character shown will be</td>
</tr>
</tbody>
</table>
substituted (, or .) based on international settings.

, Thousands placeholder. Actual character shown will be substituted (, or .) based on international settings.

: Time separator. Actual character shown will be substituted based on international settings.

/ Date separator. Actual character shown will be substituted based on international settings.

\ Treat next character as a literal. Allows use of placeholder characters as literals. Treated as a literal for masking purposes.

& Character placeholder. ANSI characters from 32-126 and from 128-255.

> Convert all characters that follow to uppercase.

< Convert all characters that follow to lowercase.

A Required alphanumeric character placeholder.

a Optional alphanumeric character placeholder.

9 Optional digit character placeholder.

C Optional character or space placeholder.

? Letter placeholder.

[Literal] All other characters are displayed as themselves as literals.

Format Property

The Format property modifies how the MaskedEdit control displays and prints using standard formats for numbers, currency and date/time data.

<table>
<thead>
<tr>
<th>Format Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Empty)</td>
<td>Default numeric format. Displays as entered.</td>
</tr>
<tr>
<td>$#,##0.00;($#,##0.00)</td>
<td>Currency format using thousands separator with negative numbers in parentheses.</td>
</tr>
<tr>
<td>0</td>
<td>Fixed number format displaying at least one digit.</td>
</tr>
<tr>
<td>#,##0</td>
<td>Commas format using commas as thousands separator.</td>
</tr>
<tr>
<td>0%</td>
<td>Percent format. Multiply Value by 100 and append a percent sign.</td>
</tr>
<tr>
<td>0.00E+00</td>
<td>Scientific notation.</td>
</tr>
<tr>
<td>c</td>
<td>Default general date and time format. Display date, time or both.</td>
</tr>
<tr>
<td>ddddd</td>
<td>Long date format taken from International setting of Control Panel.</td>
</tr>
<tr>
<td>dd-mmm-yy</td>
<td>Medium date format.</td>
</tr>
<tr>
<td>ddddd</td>
<td>Short date format taken from International setting of Control Panel.</td>
</tr>
</tbody>
</table>
Long time format taken from International setting of Control Panel.

Medium time format.

Short time format.

Tips

The MaskedEdit control is a bound control, so it can be linked to a data control and thus display field values for the current record in a data set. It can also write values to a data set. Use the three bound properties of MaskedEdit to do this: DataChanged, DataField and DataSource. See your favorite reference on the ADO Data Control for more info on such uses.
Control - Rich Text Box

Allows entry and editing of text using advanced formatting features, such as paragraph formatting with left and right indents, hanging indents, bold and italic font and so on. Also supports object embedding using the OLEObjects collection.

RichTextBox does not have the 64K character limit of ordinary TextBox controls, plus it also handles drag and drop from both the clipboard and OLE with mouse button behavior like Microsoft Word. In just a few weekends of work, you too can write your very own word processor using this control in a Manifold script.

This control has a large number of run time properties that do not appear in the Properties dialog. These are listed in this topic together with a summary list of methods to show the great range covered by a "big" control. They should be used based on guidance from a comprehensive text on Visual Basic programming that includes a detailed reference for this control.

Note: Non-Common Control properties are prefaced with "rtf", suppressed in the following for greater legibility.

Properties

In addition to the properties listed below, a Control tab provides access to certain frequently used form properties for this control. See the Form Properties topic for information on those properties.

RichTextBox

General

FileName [load from] The filename of the .txt or .rtf file loaded into the rich text box.

OLEDragMode DragManual (default) or DragAutomatic. DragManual means the program handles all OLE drag/drop operations, while DragAutomatic means the rich text box handles all OLE drag/drop operations.

OLEDropMode Set to None (default) if the rich text box does not accept OLE drops and is to display the No Drop cursor if an OLE drop is attempted. Set to Manual, the rich text box will trigger OLE drop events, allowing programs to handle the OLE drop operation in code.

RightMargin An integer specifying the indent in twips from the right edge of the text to the right edge of the control. Used for text wrapping, centering and indentation. Default 0 wraps to rightmost part of the control. Take into account border widths if using FixedSingle is set in BorderStyle.

MaxLength Long integer specifying the maximum number of characters a user can enter. Default is zero, indicating no maximum.

BulletIndent If SetBullet is True, indicates the amount of indent in dialog based units. Will return NULL if the selection spans multiple paragraphs with different margin settings.

AutoVerbMenu Determines if a pop-up menu containing a selected object's verbs is displayed when the user right-clicks an OLE object. True - Menu displayed. False - No menu. When True, Click and MouseDown events won't occur when the OLE object is right clicked.

Enabled True or False. True means the object can respond to user-generated events, false prevents it from responding. We can disable a control like a rich text box if we wish it to simply display information in a read-only way.

Locked True or False. True means the user can scroll and highlight the text in the box but cannot change it. The


program can change the text by modifying the rich text box's **Text** property. False (default) means users can edit the text in the rich text box.

**MultiLine** True or False. True (checked) enables multiple lines of text. A multiple line rich text box will wrap text as the user enters text that extends past the rich text box. If a horizontal scroll bar is present text will not wrap. On a form with no default button, pressing ENTER in a multiple line rich text box will move the focus to the next line. If a default button exists in the form, the user must press CTRL Enter to move to the next line.

**HideSelection** True or False. True means that highlighted text will not appear highlighted when the rich text box loses the focus. False means that highlighted text stays highlighted even when another form or dialog has the focus.

**Appearance**

**Appearance** 3D or Flat. 3D is the standard Windows look. Flat results in a rectangular square of background color with a foreground color border.

**BorderStyle** FixedSingle (default) or None. None may be used with Flat **Appearance** to achieve an uncaptioned rectangle of background color.

**MousePointer** The type of mouse pointer displayed when over the rich text box. Choose from 16 different types plus a custom pointer.

**ScrollBars** NoScrollbars, Horizontal, Vertical or Both. Set as desired to cause scroll bars to appear. Scroll bars appear only if **MultiLine** is set True (checked). Scroll bars will always appear in rich text boxes if enabled by this property even if the text does not extend beyond the borders of the box.

**DisableNoScroll** Determines if scroll bars are disabled. False (Default) - Scroll bars appear normally when displayed. True - scroll bars appear dimmed when displayed and are not enabled. Ignored when ScrollBars property is set to zero, and otherwise scroll bars appear dimmed when lines are too short or lines too few to require scrolling.

**Color**

**Back Color** Color to be used for the rich text box's background.

**Color Set** Choose from Standard Colors (standard Windows non-dithered colors) or Windows System Colors. The latter will be defined by the user's Control Panel settings and is normally the setting used so that the form changes appearance like the rest of Windows if the Control Panel settings are changed.

**Color Palette** Displays available colors. Click on the property to be changed to highlight it in the **Properties** pane, click on the desired color in the **Color Palette** pane and then press **Apply**.

**Edit Custom Color** Change the custom color presented in the **Color Palette** when the **Color Set** is set to Windows System Colors.

**Font**
Properties
Font properties that may be changes.

Font
Choose a font installed on this system. It's wise to choose standard Windows fonts such as MS Sans Serif that are universally available.

Size
Size of font, in points.

Effects
Bold, Italic, Underline or Strikeout.

Sample Text
A preview of the selected settings.

Picture

Mouselcon
Custom icon to use when the MousePointer property is set to 99. Like Visual Basic, will not load animated cursor (.ani) files.

Note: The RichTextBox is a data-bound control, so we can bind it with a Data control to a large capacity field in a database system (for example, a TEXT type field in SQL Server or a Binary or Memo field in Access).

Properties
Following is a listing of properties for the RichTextBox control that may be used at run time. We can exploit these properties to create a full-power, professional word processor if so desired.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SelHangingIndent</td>
<td>Return or set integer margins for the paragraph(s).</td>
</tr>
<tr>
<td>SelIndent</td>
<td></td>
</tr>
<tr>
<td>SelRightIndent</td>
<td></td>
</tr>
<tr>
<td>SelAlignment</td>
<td>Controls alignment of paragraph(s). rtfLeft, rtfRight, rtfCenter or Null (current selection spans paragraphs with different alignments).</td>
</tr>
<tr>
<td>SelBold</td>
<td>Return or set font style of currently selected text. False (Default) - None of the characters in the selection or character following the insertion point have the font style. True - All characters have the font style. Null - A mix of font styles.</td>
</tr>
<tr>
<td>SelItalic</td>
<td></td>
</tr>
<tr>
<td>SelStrikethur</td>
<td></td>
</tr>
<tr>
<td>SelUnderline</td>
<td></td>
</tr>
<tr>
<td>SelCharOffset</td>
<td>Integer: Whether character(s) are on the normal text baseline. 0 - character(s) all are on baseline. Null - mix of offsets. &gt; 0 - Superscript above the baseline by given number in twips. &lt; 0 - Subscript below the baseline by given number in twips.</td>
</tr>
<tr>
<td>SelColor</td>
<td>Returns or sets value that determines color of selected text. The value specifies RGB colors in code using RGB or QBColor functions, or System color constants. Null - a mix of colors in selected text.</td>
</tr>
<tr>
<td>SelFontName</td>
<td>Returns or sets font for currently selected text or character(s) immediately following insertion point, using a string that identifies a font installed on the the system.</td>
</tr>
</tbody>
</table>
system. Returns Null if text contains a mix of fonts.

SelFontSize
Returns or sets font size for currently selected text or character(s) immediately following insertion point, using an integer for font size in points up to 2160. Returns Null if text contains a mix of sizes.

Set the SelFontName first, then the size and style.

SelBullet
Returns or sets value (integer or constant) determining bullet style of paragraph containing current selection or insertion point. False (Default) - paragraph(s) don’t have the bullet style. True - paragraph(s) have bullet style. Null - paragraph(s) have a mix of bullet and non-bullet styles.

SelTabCount
The number of tab positions in the selected paragraph(s) or in those paragraph(s) following insertion point.

SelTabs(index)
Returns or sets location for a designated tab, where index is an integer beginning at zero that identifies a specific tab. Location is an integer in dialog units.

SelRTF
Returns or sets the text string (in .rtf format) in the current selection. Returns zero length string if no text selected. Used with Print function to write .rtf files.

TextRTF
Returns or sets text in a RichTextBox control, including all .rtf code. Setting this property replaces entire contents with the new string. Used with Print function to write .rtf files.

SelProtected
Determines if current selection is protected. True - All characters are protected. False - None are protected. Null - A mix of protected/unprotected characters in the selection.

Protected text cannot be modified by the user. Used to create forms including areas of fixed text.

OLEObjects
Returns a reference to the OLEObjects collection.

DataMember
Returns or sets a specified data member from those offered by the data provider.

DataFormat
Returns or sets the StdDataFormat object to which a bound object is attached. [Available at design time.]

DataBindings
Returns DataBindings collection object containing the bindable properties available to the programmer.

TabIndex
Returns or sets the tab order of objects within the Form.

DragIcon
Returns or sets icon to be used as the pointer in a drag and drop operation.

DragMode
0 - Set VbManual (default) drag and drop mode. Requires using Drag method to initiate a drag and drop. 1 - Set VbAutomatic mode to automatically initiate a drag and drop.

CausesValidation
Determines whether Validation event will occur on a second control from which the focus is being shifted. True (Default) - control from which focus shifts causes Validate event. False - control will not cause its Validate event.

TabStop
Whether the user can use the TAB key to move focus. True (Default) - designates object as a tab stop. False - bypasses the object when user is tabbing.

HelpContextID
Returns or sets positive integer specifying the context number of the Help topic for this object. 0 (Default) -
no context number specified. You will be writing context-sensitive Help for your Manifold script dialogs... right?

<table>
<thead>
<tr>
<th>Name</th>
<th>Returns the name used in code to identify this control.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parent</td>
<td>Returns the Form that contains this control.</td>
</tr>
<tr>
<td>Font</td>
<td>Returns a Font object. Used to identify a specific Font object whose properties we wish to use.</td>
</tr>
<tr>
<td>Container</td>
<td>Always returns the Form, since there are no other containers available in Manifold script forms.</td>
</tr>
<tr>
<td>ToolTipText</td>
<td>Returns or sets the string to use as a tool tip for this control. Dynamically modified tool tips, anyone?</td>
</tr>
<tr>
<td>DataChanged</td>
<td>Indicates if data in this bound control has been changed by some process other than retrieving data from the current record. False (Default) - data currently in the control, if any, is the same as in the current record. True - it's not.</td>
</tr>
<tr>
<td>DataField</td>
<td>The name of a field to which this data consumer is bound.</td>
</tr>
<tr>
<td>DataSource</td>
<td>The data source through which this data consumer is bound to a database.</td>
</tr>
<tr>
<td>WhatsThisHelpID</td>
<td>An integer for the context number of context-sensitive Help for Windows '95 pop-up help.</td>
</tr>
<tr>
<td>SelLength</td>
<td>The number of characters selected.</td>
</tr>
<tr>
<td>SelStart</td>
<td>An index, the starting point of text selected or the position of the insertion point of no text selected.</td>
</tr>
<tr>
<td>SelText</td>
<td>A string containing the currently selected text or zero length string if no text selected.</td>
</tr>
<tr>
<td>Text</td>
<td>A string containing the text in the RichTextBox.</td>
</tr>
<tr>
<td>Index</td>
<td>A number that uniquely identifies this object.</td>
</tr>
<tr>
<td>Tag</td>
<td>A string that stores any extra data we like about our object.</td>
</tr>
<tr>
<td>Visible</td>
<td>True (Default) - this object is visible. False - this object is hidden.</td>
</tr>
<tr>
<td>Object</td>
<td>Returns the object, object's property or object's method. Used to specify an object we want to use in an Automation task.</td>
</tr>
</tbody>
</table>

Methods

Following is a short summary of methods available with the RichTextBox control.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Find</strong></td>
<td>Searches the text for a given string using one or more options, as follow.</td>
</tr>
<tr>
<td>string, start, end, options</td>
<td>rtfWholeWord&lt;br&gt;rtfMatchCase</td>
</tr>
</tbody>
</table>

2477
rtfNoHighlight

**GetLineFromChar**

Parameters:
- `charpos`: long integer character position index number.

Returns number of the line containing a specified long integer character position index number.

**LoadFile**

Parameters:
- `pathname`: string
- `filetype`: string

Loads a file using following file types:
- `rtfRTF`
- `rtfText`

**SaveFile**

Parameters:
- `pathname`: string
- `filetype`: string

Saves contents to a file using following file types:
- `rtfRTF`
- `rtfText`

**SelPrint**

Parameters:
- `hdc`: device context

Sends formatted text to a device for printing using the device context for the target printing device.

Example:

```vbnet
RichTextBox1.SelPrint(Printer.hDC)
```

**Span**

Parameters:
- `characterset`: string
- `forward`: boolean
- `negate`: boolean

Select text based on specified characterset.

**Upto**

Parameters:
- `characterset`: string
- `forward`: boolean
- `negate`: boolean

Moves the insertion point up to but not including the first character that is a member of the specified characterset.

**SetFocus**

Moves the focus to the specified form or control.

**Drag**

Parameters:
- `action`: string

Begins, ends or cancels a drag operation using optional action as follows:
- `vbCancel`
- `vbBeginDrag`
- `vbEndDrag`

**Move**

Parameters:
- `left`, `top`, `width`, `height`: numeric

Moves the control. Only the left argument is required.

**ZOrder**

Parameters:
- `position`: numeric

Moves the control's position relative to other instances of the same control in z-order.

**ShowWhatsThis**

Displays a selected topic in a help file using the
What's This Windows '95 popup help.

**OLEDrag** Initiates an OLE drag/drop operation.

**Refresh** Forces a complete repaint of the control.
**Control - Slider**

Insert a slider bar control that can be moved left and right. Adjust the slider by dragging it or by clicking on either side of the slider. Tick marks are optional, and the slider may be oriented horizontally or vertically.

**Note:** Non-Common Control properties are prefaced with "slid", suppressed in the following for greater legibility.

In addition to the properties listed below, a **Control tab** provides access to certain frequently used form properties for this control. See the Form Properties topic for information on those properties.

## Slider

### General

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Min</strong></td>
<td>Minimum value returned by the slider, from zero (default).</td>
</tr>
<tr>
<td><strong>Max</strong></td>
<td>Maximum value returned by the slider, typically to 10 (default) or 100.</td>
</tr>
<tr>
<td><strong>SmallChange</strong></td>
<td>The amount of change in slider value when user presses left or right arrow keys on the keyboard.</td>
</tr>
<tr>
<td><strong>LargeChange</strong></td>
<td>The amount of change in slider value when user clicks to either side of slider handle, or when user presses Page Up or Page Down on keyboard.</td>
</tr>
<tr>
<td><strong>OLEDropMode</strong></td>
<td>Set to None (default) if the slider does not accept OLE drops and is to display the No Drop cursor if an OLE drop is attempted. Set to Manual, the slider will trigger OLE drop events, allowing programs to handle the OLE drop operation in code.</td>
</tr>
<tr>
<td><strong>Enabled</strong></td>
<td>True or False. True means the object can respond to user-generated events, false prevents it from responding.</td>
</tr>
<tr>
<td><strong>SelectRange</strong></td>
<td>True - The slider can have a selected range. False - Can't have a selected range.</td>
</tr>
<tr>
<td><strong>SelStart</strong></td>
<td>Used with <strong>SelLength</strong> to select a range of values in a Slider. Start of selected range. No effect if SelectRange is False.</td>
</tr>
<tr>
<td><strong>SelLength</strong></td>
<td>Used with <strong>SelLength</strong> to select a range of values in a Slider. Length of selected range. No effect if SelectRange is False.</td>
</tr>
</tbody>
</table>

### Appearance

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Orientation</strong></td>
<td>Horizontal (Default) or Vertical slider.</td>
</tr>
<tr>
<td><strong>TickStyle</strong></td>
<td>Tick marks display style. NoTicks; BottomRight - marks on bottom of horizontal sliders and to right of vertical sliders; TopLeft - marks on top of horizontal sliders and to left of vertical sliders; Both - both sides.</td>
</tr>
<tr>
<td><strong>TickFrequency</strong></td>
<td>Frequency of tick marks. If range is 100, and TickFrequency is 5, there will be 20 tick marks on slider.</td>
</tr>
<tr>
<td><strong>MousePointer</strong></td>
<td>The type of mouse pointer displayed when over the slider. Choose from 16 different types plus a custom pointer</td>
</tr>
<tr>
<td><strong>Text Position</strong></td>
<td>ToolTipText position, used for number readout when dragging. AboveLeft - above horizontal sliders and to left of vertical sliders. BelowRight - the other way.</td>
</tr>
<tr>
<td>MouseIcon</td>
<td>Custom icon to use when the MousePointer property is set to 99. Like Visual Basic, will not load animated cursor (.ani) files.</td>
</tr>
</tbody>
</table>
Control - Tree View

Insert a hierarchical tree display. Designed to display data such as organization trees, entries in an index, files and directories on disk, etc.

**Note:** Non-Common Control properties are prefaced with "tvw", suppressed in the following for greater legibility.

**Properties**

In addition to the properties listed below, a **Control tab** provides access to certain frequently used form properties for this control. See the Form Properties topic for information on those properties.

**TreeView**

**General**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Style</strong></td>
<td>The graphics style (images, text, plus/minus boxes and lines) used to show each Node object. Text only; image and text; +/- and text; +/-, image and text; lines and text; lines, image, and text; (Default) lines, +/-, image and text.</td>
</tr>
<tr>
<td><strong>MousePointer</strong></td>
<td>The type of mouse pointer displayed when over the tree view. Choose from 16 different types plus a custom pointer</td>
</tr>
<tr>
<td><strong>LineStyle</strong></td>
<td>TreeLines - show lines between Node siblings and their parent Node. RootLines - also show lines between root nodes.</td>
</tr>
<tr>
<td><strong>LabelEdit</strong></td>
<td>Automatic - BeforeLabelEdit event is generated when the user clicks the label of a selected node. Manual - BeforeLabelEdit event generated only when the StartLabelEdit method is invoked.</td>
</tr>
<tr>
<td><strong>ImageList</strong></td>
<td>ImageList control to use with this treeview</td>
</tr>
<tr>
<td><strong>BorderStyle</strong></td>
<td>FixedSingle (default) or None. None may be used with Flat Appearance to achieve an uncaptioned rectangle of background color.</td>
</tr>
<tr>
<td><strong>Appearance</strong></td>
<td>3D or Flat. 3D is the standard Windows look. Flat results in a rectangular square of background color with a foreground color border.</td>
</tr>
<tr>
<td><strong>OLEDragMode</strong></td>
<td>DragManual (default) or DragAutomatic. DragManual means the program handles all OLE drag/drop operations, while DragAutomatic means the tree view handles all OLE drag/drop operations</td>
</tr>
<tr>
<td><strong>OLEDropMode</strong></td>
<td>Set to None (default) if the tree view does not accept OLE drops and is to display the No Drop cursor if an OLE drop is attempted. Set to Manual, the tree view will trigger OLE drop events, allowing programs to handle the OLE drop operation in code.</td>
</tr>
<tr>
<td><strong>Indentation</strong></td>
<td>Indentation of objects in the tree view.</td>
</tr>
<tr>
<td><strong>PathSeparator</strong></td>
<td>Delimiter character used for the path returned by the FullPath property. &quot;&quot; by default.</td>
</tr>
<tr>
<td><strong>HideSelection</strong></td>
<td>True or False. True means that highlighted text will not appear highlighted when the tree view loses the focus. False means that highlighted text stays highlighted even when another form or dialog has the focus.</td>
</tr>
<tr>
<td><strong>Sorted</strong></td>
<td>True - Node objects sorted by their Text property. False - Not sorted.</td>
</tr>
</tbody>
</table>
**FullRowSelect**
Specifies if the entire row is selected.

**Enabled**
True or False. True means the object can respond to user-generated events, false prevents it from responding.

**Checkboxes**
False (Default) - Checkboxes do not appear. True - Checkboxes appear.

**SingleSel**
Whether the item is expanded when selected. False (Default) - The item doesn't expand. True - the item expands.

**Scroll**
True - scrollbars are displayed. False (Default) - no scrollbars.

**HotTracking**
Whether or not mouse-sensitive highlighting is enabled. True - Header captions are highlighted as the mouse passes over them. False - HeaderItem objects do not respond to mouse movement unless the mouse button is clicked.

**Font**

**Properties**
Font properties that may be changes.

**Font**
Choose a font installed on this system. It's wise to choose standard Windows fonts such as MS Sans Serif that are universally available.

**Size**
Size of font, in points.

**Effects**
Bold, Italic, Underline or Strikeout.

**Sample Text**
A preview of the selected settings.

**Picture**

**MouseIcon**
Custom icon to use when the **MousePointer** property is set to 99. Like Visual Basic, will not load animated cursor (.ani) files.
Control - UpDown

Insert an increment/decrement control, also called spin buttons.

UpDown controls are most often used with another control, such as a Text Box, which is known as the buddy control of the UpDown. [For non-English speakers: a "buddy" is slang for a "friend."\]

In addition to the properties listed below, a Control tab provides access to certain frequently used form properties for this control. See the Form Properties topic for information on those properties.

UpDown

General

Alignment
- AlignmentLeft - Place on left side of buddy control.
- AlignmentRight - Place on right side of buddy control.

Orientation
- OrientationVertical - Arrows point up/down.
- OrientationHorizontal - Arrows point left/right.

OLEDropMode
Set to None (default) if the UpDown does not accept OLE drops and is to display the No Drop cursor if an OLE drop is attempted. Set to Manual, the UpDown will trigger OLE drop events, allowing programs to handle the OLE drop operation in code.

Buddy

Buddy Control
Variant value that specifies the buddy control.

AutoBuddy
True - Automatically use previous control in tab order as buddy control.

SyncBuddy
True - UpDown synchronizes the Value property with the designated property (set in BuddyProperty) of the buddy control. False (Default) - no synchronization of Value with a property in the buddy control.

BuddyProperty
Property in buddy control to synchronize to the Value property of the UpDown. If no property is specified and SyncBuddy is True, then the default property of the buddy control is used.

Scrolling

Value
Long integer specifying current value, incremented or decremented by up and down arrows.

Min
Minimum value returned by the UpDown.

Max
Maximum value returned by the UpDown.

Wrap
True - wrap the value when user reaches the min or max value. False (Default) - don't wrap around.

Increment
The amount by which Value changes with each click of the up or down arrow. Default = 1.

Max is normally greater than Min. The up arrow always moves Value in the direction of Max and the down arrow always towards Min. Both can be negative numbers. If Max can be less than Min (very weird and asking for conceptual errors - not a good idea).

Note: Some texts on VB (including, surprisingly, MDSN) say that we cannot use the windowless, lightweight form of controls as a buddy control for the UpDown. That's not true, as wTextBox, for example works fine with the
UpDown both within Visual Basic and Manifold Visual Basic Scripting forms. Should Microsoft ever change this control in future releases so that windowless controls cannot be buddies, one could always simply use the UpClick and DownClick events generated by the UpDown to change the value in the text box.
Control - Tab Strip

Insert a tabbed control where clicking on each tab brings it to the fore. Also can be used to provide a toolbar look, including a cool "flat" toolbar look.

To load up a tab strip, one must create a container on the form that's loaded with the arrangement of controls to be displayed in the client area of the tab strip. At run time, as each tab is pressed we use Move to position the container full of desired controls into position into the client area.

**Note:** Non-Common Control properties are prefaced with "tab", suppressed in the following for greater legibility.

**Properties**

In addition to the properties listed below, a **Control tab** provides access to certain frequently used form properties for this control. See the Form Properties topic for information on those properties.

**Tabla**

**General**

- **MultiRow** True - Show many tabs as cascaded rows. False - add left/right scroll buttons to scroll through more tabs than can fit in the width of the control.
- **ShowTips** True (Default) - Show ToolTipsText. False - Don't.
- **Enabled** True or False. True means the object can respond to user-generated events, false prevents it from responding.
- **MousePointer** The type of mouse pointer displayed when over the tab strip. Choose from 16 different types plus a custom pointer
- **Style** Tabs (Default) - standard tabbed look. Buttons - tabs display as a row of buttons with no border around client area. Use to create a toolbar look. FlatButtons - button is flat until pressed and then takes on an indented look.
- **TabWidthStyle** Used when MultiRow is True. Justified - evens out width of tabs to evenly fill each row. NonJustified - each tab takes on the width of its contents. Fixed - use TabFixedWidth for all tabs.
- **ImageList** ImageList control to use with this treeview
- **OLEDropMode** Set to None (default) if the tab strip does not accept OLE drops and is to display the No Drop cursor if an OLE drop is attempted. Set to Manual, the tab strip will trigger OLE drop events, allowing programs to handle the OLE drop operation in code.
- **Placement** Tabs appear at Top, Bottom, Left or Right side of the control. Use with TabStyle set to TabOpposite to achieve a (weird) effect as if tabs are being moved from one side of a book to another.
- **TabStyle** Used with MultiRow = True. TabStandard (Default) - remaining tabs stay on same side of the control. TabOpposite - Rows of tabs in front of the row containing the selected tab are displayed on the opposite side of the control.
- **TabFixedWidth** Specify a fixed width to use when TabWidthStyle is set to Fixed.
- **TabFixedHeight** Specify a fixed height to use. If not set, TabStrip will use either the height of the font or the height of the images to compute the height of tabs.
### Programming

**TabMinWidth**  
The minimum allowable width of a tab.

**HotTracking**  
Whether or not mouse-sensitive highlighting is enabled.  
True - captions are highlighted as the mouse passes over them.  
False (Default) - objects do not respond to mouse movement unless the mouse button is clicked.  
True is a good idea with FlatButton to achieve a "live" toolbar look that shows a raised button on a mouse hover.

**Separators**  
True - Draw separators between buttons when Style is set to Button or FlatButton.  
False (Default) - No separators.  
Separators help users understand this is a control and not just a caption when FlatButton is used.

**MultiSelect**  
Whether or not user can make multiple selections from the tab strip.  
NoMultiSelect means none.  
Simple, means a mouse click or SPACEBAR selects/deselects while arrow keys move focus.  
Extended adds CTRL and SHIFT click selection in the usual way.

### Tabs

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Index**    | Integer that uniquely identifies a tab object in the Tab Strip collection.  
Numbered in order of creation from the first tab object, which is always 1. |
| **Caption**  | Text for tab caption. |
| **Key**      | A unique string that identifies each tab object. |
| **Tag**      | Stores a string available for the programmer to use as desired.  
Often used to store an identification string of some sort. |
| **ToolTipText** | Tool tip text to display as tool tip for this tab. |
| **Image**    | Image to use for this tab, if ImageList specified. |

### Font

**Properties**  
Font properties that may be changes.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Font**     | Choose a font installed on this system.  
It's wise to choose standard Windows fonts such as **MS Sans Serif** that are universally available. |
| **Size**     | Size of font, in points. |
| **Effects**  | Bold, Italic, Underline or Strikeout. |
| **Sample Text** | A preview of the selected settings. |

### Picture

**MouseIcon**  
Custom icon to use when the **MousePointer** property is set to **99**.  
Like Visual Basic, will not load animated cursor (.ani) files.
Control - Picture Clip

Displays graphics from bitmap, icon, metafile, JPEG or GIF files.
Control - Multimedia Control

Manages Media Control Interface (MCI) devices like sound boards, MIDI sequencers, CD-ROM drives, audio players, videodisc players, etc. Add voice note recording to your dialog, or play spoken announcements.

This control provides push buttons that issue MCI commands to supported devices. The application should have the MCI device open before the user is allowed to push buttons. When recording audio, open a new file to be sure that the data file is in a format compatible with the system’s recording capability. Issue the MCI Save command before closing the device to store the recording in the file.

Note: Non-Common Control properties are prefaced with "mci", suppressed in the following for greater legibility.

Properties

In addition to the properties listed below, a Control tab provides access to certain frequently used form properties for this control. See the Form Properties topic for information on those properties.

MMControl (Multimedia MCI Control)

General

- **DeviceType**: A string specifying type of MCI device to open: AVIVideo, CDAudio, DAT, DigitalVideo, MMMovie, Other, Overlay, Scanner, Sequencer, VCR, Videodisc or WaveAudio.
- **FileName**: file to be opened by Open or saved by Save. To change FileName at run time, must close and reopen the control.
- **MousePointer**: The type of mouse pointer displayed when over the Multimedia Control. Choose from 16 different types plus a custom pointer.
- **Orientation**: Horizontal (Default) or Vertical Multimedia Control.
- **BorderStyle**: FixedSingle (default) or None. None may be used with Flat Appearance to achieve an uncaptioned rectangle of background color.
- **RecordMode**: Either RecordInsert or RecordOverwrite. Must try and see with a device. WaveAudio devices support Insert mode only.
- **OLEDropMode**: Set to None (default) if the Multimedia Control does not accept OLE drops and is to display the No Drop cursor if an OLE drop is attempted. Set to Manual, the Multimedia Control will trigger OLE drop events, allowing programs to handle the OLE drop operation in code.
- **UpdateInterval**: The number of milliseconds between successive StatusUpdate events.
- **Frames**: The number of frames Step or Back commands will move forward or backward.
- **Silent**: True - Sound is turned off. False - any sound is played.
- **Shareable**: True - more than one control or application can open this device. False - only one at a time.
- **AutoEnable**: True (Default) - enable/disable buttons whose functions make sense or not given the device’s capabilities. False - buttons set by properties specified. True only works if the button is otherwise enabled in controls properties.
- **Enabled**: True or False. True means the object can respond to user-generated events, false prevents it from responding.
### Controls

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>button</td>
<td>Prev, Next, Step, Back, Pause, Play, Record, Stop or Eject.</td>
</tr>
<tr>
<td>button Enabled</td>
<td>True - enable this button. False - button always disabled.</td>
</tr>
<tr>
<td>button Visible</td>
<td>True - show this button. False - hide this button.</td>
</tr>
</tbody>
</table>

### Picture

| MouseIcon       | Custom icon to use when the MousePointer property is set to 99. Like Visual Basic, will not load animated cursor (.ani) files. |
Programming

Control - Status Bar

Provides a status bar like those at the bottom of many program windows. May be divided up into up to sixteen Panel objects, contained in a Panels collection. Panel objects can contain text and/or a picture and can automatically display frequently desired information such as time, date, etc.

**Note:** Non-Common Control properties are prefaced with "sbr", suppressed in the following for greater legibility.

**Properties**

In addition to the properties listed below, a **Control tab** provides access to certain frequently used form properties for this control. See the Form Properties topic for information on those properties.

**StatusBar**

**General**

- **Style**
  - Normal - Show all Panel objects. Simple - display one large panel.

- **MousePointer**
  - The type of mouse pointer displayed when over the status bar. Choose from 16 different types plus a custom pointer

- **SimpleText**
  - Text to show when Style is set to Simple.

- **OLEDropMode**
  - Set to None (default) if the status bar does not accept OLE drops and is to display the No Drop cursor if an OLE drop is attempted. Set to Manual, the status bar will trigger OLE drop events, allowing programs to handle the OLE drop operation in code.

- **Enabled**
  - True or False. True means the control can respond to user-generated events, false prevents it from responding.

- **ShowTips**
  - True (Default) - Show ToolTipsText. False - Don't.

**Panels**

- **Index**
  - Integer that uniquely identifies a panel object in the Status bar collection. Numbered in order of creation from the first panel object, which is always 1.

- **Text**
  - Text for panel caption.

- **ToolTipText**
  - Tool tip text to display as tool tip for this panel.

- **Key**
  - A unique string that identifies each panel object.

- **Tag**
  - Stores a string available for the programmer to use as desired. Often used to store an identification string of some sort.

- **Alignment**
  - Controls alignment of text or pictures in the panel. Left, Center or Right.

- **Style**
  - Choose from Text, to display panel's Text property, or from a collection of standard capabilities showing keyboard status.
    - **Text** - Text and/or a bitmap.
    - **Caps** - Caps lock key on/off.
    - **Num** - Num lock key on/off.
    - **Ins** - Insert key enabled on/off.
    - **Scrl** - Scroll lock key on/off.
Time - Show time in system format.
Date - Show date in system format.
Kana - Kana lock on/off [Japanese o/s only].
Bevel NoBevel - Flat text. Inset (Default) - Panel is sunk into status bar. Raised - Panel is raised up.
AutoSize What to do when container (form) resized. NoAutoSize - Width of panel specified by Width property. Spring - All panels resize to new space. [use on middle panels] Contents - Panel resizes to fit contents.
MinimumWidth Panel cannot shrink to less than this width.
ActualWidth Reports actual width of panel
Picture Bitmap to use for this panel.
Enabled True or False. True means the panel can respond to user-generated events, false prevents it from responding.
Visible True - Panel is visible. False - It isn't.

Font
Properties Font properties that may be changes.
Font Choose a font installed on this system. It's wise to choose standard Windows fonts such as MS Sans Serif that are universally available.
Size Size of font, in points.
Effects Bold, Italic, Underline or Strikeout.
Sample Text A preview of the selected settings.

Picture
MouseIcon Custom icon to use when the MousePointer property is set to 99. Like Visual Basic, will not load animated cursor (.ani) files.
Control - Tool Bar

Creates a customizable toolbar at the top of the form that hosts buttons or other controls. Toolbars contain a collection of Button objects. Button objects created using Placeholder style can reserve space on the toolbar for other controls, like ComboBox controls.

Buttons can have text, images or both. Images are fetched from an ImageList control collection. Buttons are programmed by adding code to the ButtonClick event for each button.

Buttons may even have menus within them when the button uses Dropdown style. Each button in the Buttons collection can itself have a collection of ButtonMenu objects.

Note: Non-Common Control properties are prefaced with "tbr", suppressed in the following for greater legibility.

Properties

In addition to the properties listed below, a Control tab provides access to certain frequently used form properties for this control. See the Form Properties topic for information on those properties.

ToolBar

General

MousePointer  The type of mouse pointer displayed when over the Toolbar. Choose from 16 different types plus a custom pointer

OLEDropMode  Set to None (default) if the Toolbar does not accept OLE drops and is to display the No Drop cursor if an OLE drop is attempted. Set to Manual, the Toolbar will trigger OLE drop events, allowing programs to handle the OLE drop operation in code.

ImageList  ImageList to use for buttons in normal mode.

DisabledImageList  ImageList to use for disabled buttons.

HotImageList  ImageList for "hot" images: appear on button on mouse hover when Style is Transparent.

HelpContextID  Context ID for this control to use for this toolbar.

ButtonHeight  Height of button. Updated to accommodate string in Caption property or image in Image property for the button.

ButtonWidth  Width of button. Updated to accommodate string in Caption property or image in Image property for the button.

Appearance  3D or Flat. 3D is the standard Windows look. Flat has no effect with command buttons.

Text Alignment  Position of text relative to button. TextAlignbottom or TextAlignRight

Style  Standard or Flat, the appearance of the toolbar control as either a "3D" look or as a flat look.

HelpFile  A filename for path and filename of the help file to use with this toolbar.

AllowCustomize  True - allow user to customize toolbar by double clicking on it. False - no user customization allowed.

ShowTips  True (Default) - Show ToolTipsText. False - Don't.
Wrappable  True - buttons wrap if form resized. False - No wrapping.

Enabled  True or False. True means the object can respond to user-generated events, false prevents it from responding.

BorderStyle  FixedSingle (default) or None.

Buttons

Index  Integer that uniquely identifies a button object in the Toolbar collection. Numbered in order of creation from the first button object, which is always 1.

Caption  Text for button caption.

Key  A unique string that identifies each button object.

Style  Specify button behavior:

  Default  Standard toolbar button click and it bounces back.

  Check  Used for toggles/mode buttons. Pushed in and stays in until pushed out.

ButtonGroup  Allows only one button in the group to be pushed in.

Separator  No function. A spacer of eight pixels.

Placeholder  Reserves space into which other controls can be placed.

Dropdown  Enables use of button menu on this button. Creates a small down arrow to indicate the presence of a dropdown menu.

ToolTipText  Tool tip text to display as tool tip for this button.

Tag  Stores a string available for the programmer to use as desired. Often used to store an identification string of some sort.

Visible  True - Button is visible. False - It isn't.

Enabled  True or False. True means the button can respond to user-generated events, false prevents it from responding.

MixedState  Dithers the images on the button to be used to display indeterminate states associated with that button.

Description  Description that will appear for this button when the Customize Toolbar dialog is invoked.

Value  Pressed or Unpressed.

Width (Placeholder)  Width to use for placeholder Style buttons.

Image  Image number from ImageList.

ButtonMenus

Index  Integer that uniquely identifies a ButtonMenu object in the button menu collection. Numbered in order of creation from the first ButtonMenu object, which is always 1.

Text  Text for menu item.

Key  A unique string that identifies each ButtonMenu object.
**Tag**  Stores a string available for the programmer to use as desired. Often used to store an identification string of some sort.

**Enabled**  True or False. True means the button can respond to user-generated events, false prevents it from responding.

**Visible**  True - Button is visible. False - It isn't.

**Picture**

**MouseIcon**  Custom icon to use when the MousePointer property is set to 99. Like Visual Basic, will not load animated cursor (.ani) files.
Control - Cool Bar

Provides a modern, "railbar" look and the ability to create user configurable toolbars like those in IE. Requires installation of Internet Explorer 3.0 or greater on both the development machine and the run-time machine.

The CoolBar is a container that can host child controls; however, a CoolBar can only host controls that expose a window handle (so, no windowless lightweight controls as are found on the frequently used Tools toolbar in Manifold). The resizable regions in a CoolBar are called bands. Each band can host a child control.

Properties

In addition to the properties listed below, a Control tab provides access to certain frequently used form properties for this control. See the Form Properties topic for information on those properties.

CoolBar

General

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ImageList</td>
<td>ImageList to use for the CoolBar.</td>
</tr>
<tr>
<td>Orientation</td>
<td>Vertical or Horizontal</td>
</tr>
<tr>
<td>Picture</td>
<td>Image to use to &quot;tile&quot; CoolBar. Calls Picture and Color (see below) properties dialog.</td>
</tr>
<tr>
<td>EmbossPicture</td>
<td>Convert picture to monochrome 3D look.</td>
</tr>
<tr>
<td>EmbossHighlight</td>
<td>Color to use for embossing highlights. Available when EmbossPicture is used.</td>
</tr>
<tr>
<td>EmbossShadow</td>
<td>Color to use for embossing shadows. Available when EmbossPicture is used.</td>
</tr>
<tr>
<td>ForeColor</td>
<td>Foreground color (text).</td>
</tr>
<tr>
<td>BackColor</td>
<td>Background color of coolbar.</td>
</tr>
<tr>
<td>BandBorders</td>
<td>True - Apply 3D borders between band rows.</td>
</tr>
<tr>
<td>FixedOrder</td>
<td>True - User cannot rearrange bands at run time. False - User can rearrange bands.</td>
</tr>
<tr>
<td>VariantHeight</td>
<td>True (Default) - Band heights may vary. False - Band heights are all the same.</td>
</tr>
</tbody>
</table>

Bands

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index</td>
<td>Integer that uniquely identifies a band object in the CoolBar collection. Numbered in order of creation from the first band object, which is always 1.</td>
</tr>
<tr>
<td>Child</td>
<td>Loaded with controls drawn on this CoolBar. Choose a control from the Child list and press Apply to associate it with this band.</td>
</tr>
<tr>
<td>Style</td>
<td>BandNormal (Default) Band can be resized. BandFixedSize Band can not be resized.</td>
</tr>
<tr>
<td>UseCoolBarPicture</td>
<td>Tile this band with the coolbar picture.</td>
</tr>
<tr>
<td>EmbossPicture</td>
<td>Show picture in two colors. Available if UseCoolBarPicture is not checked.</td>
</tr>
<tr>
<td>EmbossHighlight</td>
<td>Color to use for embossing highlights. Available when EmbossPicture is used.</td>
</tr>
</tbody>
</table>
EmbossShadow  Color to use for embossing shadows.  Available when EmbossPicture is used.

UseCoolbarColors  Use the same colors as defined for CoolBar

ForeColor  Foreground color.  Available when UseCoolbarColors is not checked.

BackColor  Background color.  Available when UseCoolbarColors is not checked.

Caption  Text for band caption.

Width  Width of this band.

MinWidth  Minimum width allowed during resizing.

MinHeight  Minimum height allowed during resizing.

Key  A unique string that identifies each band object.

Tag  Stores a string available for the programmer to use as desired.  Often used to store an identification string of some sort.

Image  Image from ImageList to use for this band.  Image will be displayed at run time between the move handle and the caption.

Visible  True - Band is visible.  False - It isn't.

AllowVertical  True (Default) - This band will be displayed if CoolBar in vertical orientation.  False - Will not be displayed when vertical.

NewRow  Place on new row.

FixedBackground  True (Default) - background picture is fixed on resize.  False - retile background picture on resize.

Color

Back Color  Color to be used for the band's background.

EmbossHighlight  Color to use for "light" color.

EmbossShadow  Color to use for "dark" color.

ForeColor  Color to be used for the band's foreground, the color of the text caption.

Color Set  Choose from Standard Colors (standard Windows non-dithered colors) or Windows System Colors. The latter will be defined by the user's Control Panel settings and is normally the setting used so that the form changes appearance like the rest of Windows if the Control Panel settings are changed.

Color Palette  Displays available colors.  Click on the property to be changed to highlight it in the Properties pane, click on the desired color in the Color Palette pane and then press Apply.

Edit Custom Color  Change the custom color presented in the Color Palette when the Color Set is set to Windows System Colors.
Control - Common Dialog Control

Provides a standard set of Windows dialog boxes for opening and saving files and selecting colors and fonts, and printing. Can also display Help. Invisible at run time until called.

Properties

In addition to the properties listed below, a Control tab provides access to certain frequently used form properties for this control. See the Form Properties topic for information on those properties.

CommonDialog

Open / Save As

Displayed by ShowOpen and ShowSave methods.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DialogTitle</td>
<td>Name to display in the title bar.</td>
</tr>
<tr>
<td>FileName</td>
<td>Initial path and file name to use, if desired.</td>
</tr>
<tr>
<td>InitDir</td>
<td>Initial directory to use. If not present, current directory will be used.</td>
</tr>
<tr>
<td>Filter</td>
<td>String specifying the file extension. * .txt specifies all text files. Can specify more than one type.</td>
</tr>
<tr>
<td>CancelError</td>
<td>False (Default) - no error generated when user chooses the Cancel button. True - error is generated.</td>
</tr>
<tr>
<td>Flags</td>
<td>A numeric value that is a combination of numeric values for various flags (listed in purple color below) that are summed to create the Flags value. See Microsoft's MSDN documentation for numeric values for the following flags.</td>
</tr>
<tr>
<td>AllowMultiselect</td>
<td>User can select more than one file.</td>
</tr>
<tr>
<td>CreatePrompt</td>
<td>Prompts user to create file that does not exist.</td>
</tr>
<tr>
<td>Explorer</td>
<td>For '95 and 4.0: use Explorer-like Open A File dialog box template.</td>
</tr>
<tr>
<td>ExtensionDifferent</td>
<td>Indicates extension of returned filename is different from extension specified by DefaultExt.</td>
</tr>
<tr>
<td>FileMustExist</td>
<td>User must enter name of existing file.</td>
</tr>
<tr>
<td>HelpButton</td>
<td>Display help button.</td>
</tr>
<tr>
<td>HideReadOnly</td>
<td>Hides the Read Only check box.</td>
</tr>
<tr>
<td>LongNames</td>
<td>Use long file names.</td>
</tr>
<tr>
<td>NoChangeDir</td>
<td>Dialog will set current directory to what it was when dialog was opened.</td>
</tr>
<tr>
<td>NoDereferenceLinks</td>
<td>Do not dereference shortcuts (shell links).</td>
</tr>
<tr>
<td>NoLongNames</td>
<td>No long file names allowed.</td>
</tr>
<tr>
<td>NoReadOnlyReturn</td>
<td>Returned file won't have Read Only set and won't be in a write-protected directory.</td>
</tr>
<tr>
<td>NoValidate</td>
<td>Allows invalid characters in the returned filename.</td>
</tr>
<tr>
<td>OverwritePrompt</td>
<td>User must confirm to overwrite an existing file.</td>
</tr>
</tbody>
</table>
PathMustExist User can only enter valid paths.
ReadOnly Read only box checked by default.
ShareAware Sharing violation errors will be ignored.

DefaultExt Default filename extension, such as .txt or .mdb.
MaxFileSize Integer specifying the maximum size of the filename in bytes. Default is 256.
FilterIndex Specifies the default filter when using Filter property. First defined filter is 1.

Color Displayed by ShowColor method.

Color Returns or sets selected color.
Flags A numeric value that is a combination of numeric values for various flags (listed in purple color below) that are summed to create the Flags value. See Microsoft's MSDN documentation for numeric values for the following flags.
FullOpen Entire color dialog is shown.
HelpButton Display a help button.
PreventFullOpen Prevent definition of Custom Colors.
RGBInit Initial color value for dialog box.
CancelError False (Default) - no error generated when user chooses the Cancel button. True - error is generated.

Font Displayed by ShowFont method.

FontName Selected font name.
FontSize Selected font size.
Min, Max Smallest and largest font sizes.
Flags A numeric value that is a combination of numeric values for various flags (listed in purple color below) that are summed to create the Flags value. See Microsoft's MSDN documentation for numeric values for the following flags.
ANSIOnly Only those fonts that use Windows character set.
Apply Enable the Apply button.
Both List screen and printer fonts.
Effects Enable strikethrough, underline and color effects.
FixedPitchOnly Selects only fixed-pitch fonts.
ForceFontExist Allow use only of fonts that exist.
<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>HelpButton</strong></td>
<td>Show help button.</td>
</tr>
<tr>
<td><strong>LimitSize</strong></td>
<td>Select only those fonts between Min and Max sizes.</td>
</tr>
<tr>
<td><strong>NoFaceSel</strong></td>
<td>No font name selected.</td>
</tr>
<tr>
<td><strong>NoSimulations</strong></td>
<td>Doesn’t allow GDI font simulations.</td>
</tr>
<tr>
<td><strong>NoSizeSel</strong></td>
<td>No font size selected.</td>
</tr>
<tr>
<td><strong>NoStyleSel</strong></td>
<td>No style was selected.</td>
</tr>
<tr>
<td><strong>NoVectorFonts</strong></td>
<td>Dialog does not allow vector-font selections.</td>
</tr>
<tr>
<td><strong>PrinterFonts</strong></td>
<td>List only those fonts supported by the printer.</td>
</tr>
<tr>
<td><strong>ScalableOnly</strong></td>
<td>List only scalable fonts.</td>
</tr>
<tr>
<td><strong>ScreenFonts</strong></td>
<td>List only screen fonts supported by this system.</td>
</tr>
<tr>
<td><strong>TTOOnly</strong></td>
<td>List only True Type Fonts.</td>
</tr>
<tr>
<td><strong>WYSIWYG</strong></td>
<td>Allow only those fonts that are both printer and screen fonts.</td>
</tr>
<tr>
<td><strong>CancelError</strong></td>
<td>False (Default) - no error generated when user chooses the Cancel button.</td>
</tr>
<tr>
<td></td>
<td>True - error is generated.</td>
</tr>
<tr>
<td><strong>Bold, Italic, Underline, Strikethru</strong></td>
<td>Set these properties by default.</td>
</tr>
</tbody>
</table>

**Print**

- **Print**
  - Displayed by the `ShowPrinter` method.

**Copies**

- Default number of copies to print.

**Flags**

- A numeric value that is a combination of numeric values for various flags (listed in purple color below) that are summed to create the `Flags` value. See Microsoft’s MSDN documentation for numeric values for the following flags.

**AllPages**

- State of All Pages option button.

**Collate**

- State of Collage check box.

**DisablePrintToFile**

- Disable Print to File check box.

**HelpButton**

- Show help button.

**HidePrintToFile**

- Hide Print to File check box.

**NoPageNums**

- Disable Pages option button and edit control.

**NoSelection**

- Disable Selection option button.

**NoWarning**

- Don’t warn if no default printer.

**PageNums**

- State of Pages option button.

**PrintSetup**

- Show Print Setup dialog instead of Print dialog box.

**PrintToFile**

- State of Print to File check box.

**ReturnDC**

- Return device context for printer selection made.
**ReturnDefault**  Return default printer name.

**ReturnIC**  Return information context for printer selection made.

**Selection**  State of selection option button.

**UseDevModeCopies**  If printer driver doesn’t support multiple copies, will disable the Number of copies spinner in Print dialog.

**FromPage, ToPage**  Default from and to page numbers.

**Min, Max**  Smallest number allowed in From box and the largest number allowed in the To box.

**Orientation**  Portrait or Landscape

**PrinterDefault**  True - Any changes user makes via Setup button will change printer defaults in the system registry. False - Setup selections in this dialog will not change system settings.

**CancelError**  False (Default) - no error generated when user chooses the Cancel button. True - error is generated.

## Help

**HelpContext**  HelpContextID: 0 - no context number >= 0 - a valid help context number.

**HelpKey**  String identifying keyword that identifies the Help topic.

**HelpFile**  Path and filename for Help file to use.

**HelpCommand**  A numeric value that is a numeric value representing one of the following options (in purple color). See Microsoft’s MSDN documentation for numeric values for the following options.

- **Command**  Execute a Help macro.
- **Contents**  Display Help contents topic.
- **Context**  Display Help for a particular context.
- **ContextPopup**  Display in pop-up a Help topic identified by a context number.
- **ForceFile**  Ensures WinHelp displays correct Help file.
- **HelpOnHelp**  Show Help for total idiots, who need help on using Help.
- **HelpIndex**  Display index of the given Help file.
- **HelpKey**  Display Help for a particular keyword.
- **PartialKey**  Search on partial keys and show matches, if any.
- **HelpQuit**  Tell Help the given help file is no longer in use.
- **SetContents**  Specify which Contents topic is shown when user presses the F1 key.
- **SetIndex**  Used when more than one index per Help file (don't do this).
Control - System Info Control

Detects system events such as desktop resizing, resolution changes, time changes. Also provides operating system platform and version information and changes in AC/battery power status and Plug/Play hardware configuration. Used when writing applications for portable devices (GIS in the field) and for developing info for your tech support team.

There are no properties settable at design time for the SysInfo control. It is invisible at run time

Frequently Used Run Time Properties and Events

In addition to the properties listed below, a Control tab provides access to certain frequently used form properties for this control. See the Form Properties topic for information on those properties.

SysInfo

Operating System

OSPPlatform Operating system.
OSVersion Version of the operating system.
OSBuild Build number of the operating system.

Desktop

ScrollBarSize Width of scroll bar in twips.
WorkAreaHeight, WorkAreaLeft, WorkAreaTop, WorkAreaWidth Height, left edge, top edge, and width of visible desktop adjusted for the Windows taskbar.
DisplayChanged Occurs when screen resolution changes.
SysColorsChanged Occurs when system color setting changes.

Note: Plug and Play hardware events are not listed in this table.

Power

ACStatus System using AC or battery power.
BatteryFullTime Value indicating full charge life of battery.
BatteryLifePercent Percentage of battery power remaining.
BatteryLifeTime Value indicating remaining life of battery.
BatteryStatus Status of battery's charge.
PowerQuery Suspend Occurs when system power about to be suspended.
PowerResume Occurs when system comes out of suspend mode.
PowerStatus Changed Occurs when power status of system changes.
PowerSuspend Occurs immediately before the system goes into suspend mode.
Control - Insert ActiveX Control

Insert any ActiveX control available on this system. Opens a dialog that lists ActiveX controls available on this system.

Choose a module from the list and press OK. The Register Module button is for use in cases where ActiveX controls exist on our system but have not yet been registered (for example, if we simply copy over an .ocx file into our ~/System32 directory but have not run an installation script or have otherwise manually registered it). It does not hurt to Register Module if a module has already been registered.

Next, click and drag into the form where the ActiveX control is to appear. Right-click on the control to set its properties, if any are settable at design time.

Debugger

The Manifold Debugger provides an integrated debugger for use with scripting in Manifold System running in 32-bit mode. The debugger is not available when Manifold is running in 64-bit mode in 64-bit Windows systems, nor at the present writing is it available in Vista. The Manifold debugger requires installation of Microsoft facilities that support debugging of ActiveX scripts (see below), which are not available in 64-bit mode in Windows. See the 32-bit and 64-bit Manifold Editions topic when running 64-bit Manifold System editions.

The Manifold debugger does not work with .NET languages (which, in addition to ActiveX languages also may be used for scripting within Manifold). .NET languages have their own debugging framework. Attempting to enter a .NET script under the debugger will display an error message.

The debugger works with VBScript, JScript and any other ActiveX scripting engine that supports Microsoft debugging calls. Note that at the present writing the ActiveState PERL and Python scripting languages do not support Microsoft debugging calls and so may not be used with the debugger even though they may be used for scripting.

Likewise, scripts written in .NET languages such as C#, VB .NET or JScript .NET are not supported by the debugger and should be debugged using the Microsoft console debugger that is part of the free .NET SDK download. Debugger commands, including the project pane's Run under Debugger button, will automatically be disabled if a script window uses .NET languages.

The debugger works together with the Call Stack, Variables and Watches panes and is controlled by the Tools toolbar configured with debugger commands when a script is open. The Edit - Breakpoints command is used to set breakpoints for use with the debugger.

Scripts associated with Active Columns can also run in the debugger. To debug an Active Column script, right click on an Active Column cell and choose Run under Debugger from the pop up menu. Doing this runs the
function, which will generate values for the active column in the context of the clicked cell. All regular runs of Active Column functions are done without the debugger.

If the Debugger is installed, when a script has the focus the tools toolbar will have debugger commands enabled. The debugger helps us develop and debug scripts in the scripting language of our choice. Commands will be enabled in the debugger toolbar as make sense given the execution state of the script.

- **Run** Run a script.
- **Run under Debugger** Run a script under control of the debugger.
- **Pause** Pause execution of a script.
- **Stop** Stop execution of a script.
- **Step Over** Step over a function.
- **Step Into** Step into a function.
- **Step Out** Step out of a function.

**What is a Debugger?**

A debugger is a specialized execution environment within which programs (like Manifold scripts) may be run in a controlled way that helps debug those programs. When a script is running under the control of a debugger it may be started and stopped, paused, re-started at a different location in the code, automatically stopped whenever it reaches a marked breakpoint and so on. At any time when a program is paused or stopped we can see what are the values of any variables or other information we would like to know.

To understand what a debugger does, let's take a moment to consider the old-fashioned way of writing and debugging programs. Traditionally, we can write a program that consists of one or more functions. If it works the first time without any flaws we're somewhat surprised but happy. More frequently a new program of any complexity will not work quite right the first time or it will exhibit unexpected behavior (a "bug") when it functions. To track down any problems we have to know what is going on inside the program as it executes.

The traditional way of finding out what a program is doing while it is executing is to insert additional lines of code for debugging purposes. For example, if we are not sure that the program reaches a particular function correctly we might insert a few lines of code that when that function is reached will print out some text like "Got to AddNextRecord() OK" or which might print out the values of variables that we suspect are being mishandled with printed output like "Executing LoopRecord with RecordNum = 36".

As any professional programmer knows, it's really tedious to lard up one's code with lots of extra debugging statements. Adding extra statements is slow and there is always the risk of forgetting some debugging statements within production code or of inadvertently changing the execution of a program by the debugging code.

A much faster way to debug code is to use a debugger like the Manifold debugger. The Manifold debugger takes advantage of features built into Microsoft's Active X scripting environment to provide sophisticated, professional-class debugging capability.

To stop the execution of a program at any given location, insert a **breakpoint** at that spot using the Edit - Breakpoints command. Whenever program flow reaches a breakpoint the program will pause at that point. We can then see the contents of any variables using the Variables pane. If we like, we can even change the content of any variable by changing it's value in the the Variables pane.

The tools toolbar allows us to control program flow within the script. Using the Call Stack pane we can even jump to different parts of the script. The Watches pane allows us to see variables, process parameters and even computed expressions. All together the ensemble of debugger tools toolbar commands and the Call Stack, Variables and Watches panes allow us to control and examine the execution of a script with much greater convenience and flexibility than is possible using traditional methods.
**Debugger Tutorial**

The following sequence of steps shows operation of the debugger together with Call Stack, Variables and Watches panes.

1. Launch Manifold, open a new project and create two scripts. The first script, called **A**, is as follows:

```
Sub Main
    Set components = Application.ActiveDocument.ComponentSet
    Set anotherScript = components(components.ItemByNames(“B”))
    anotherScript.Run
End Sub
```

The second script, called **B**, is:

```
Sub SayHello
    str = “Hello!”
    Application.MessageBox str
End Sub

Sub Main
    Call SayHello
End Sub
```

2. Using **Edit - Breakpoints** place breakpoints in the scripts as shown above.

3. Try running script **A** or **B** from the project pane. Verify that both scripts display a “Hello!” message.

4. Turn on the Call Stack, Variables and Watches panes and dock them so that all three panes are visible.

5. Open script **A** and run it under the debugger using the toolbar button in the Tools toolbar or the **Script - Run under debugger** menu command.

6. The script will stop at the breakpoint in the fourth line of script **A**. The Call Stack pane shows that we’re within “Main” at line 4. The Variables pane shows “components” and “anotherScript” variables. The Watches pane is empty.

7. In the Variables pane, expand the **components** entry. The **Count** item shows that component set consists of two components (script **A** and script **B**). The other two items variables are internal to Manifold and may be replaced in future editions with a more sensible list of components. Expand the **anotherScript** item. The **Name** item shows the name of the script component. Other items show the script language, script text and various other properties. Press in the **Show Common Properties** button in the the pane toolbar. This shows the **Application** and **Parent** properties that are common to all Manifold objects. Note that we can expand **components** and then expand its **Application** property, then expand the Application’s **ActiveDocument** property and so on. Press out the **Show Type** button within the pane toolbar. This will hide the **Type** column so there will be more room for the **Name** and **Value** columns.

8. In the Watches pane, click the **New Watch** button and enter “components”. This will display the contents of the **components** variable in a manner similar to the Variables pane. Click the **New Watch** button once more and enter “components.Count + 1”. This will display “3”. Double-click the empty line below the list of watches (an
alternative way to create a new watch) and enter "Application". This will display the built-in Application object. Click on the first watch to highlight it and click the Delete Watch button within the toolbar. This will delete the selected watch. Note that the Delete Watch button won’t work on a subitem. For example, it won’t work on the Application.ActiveDocument item within the Application watch. The Show Common Properties and Show Type buttons operate do the analogous buttons in the Variables pane toolbar.

9. Click the Step Into button within the Tools toolbar or choose the equivalent command in the Script menu. This will step into script B.

```
Sub SayHello
    str = "Hello!"
    Application.MessageBox str
End Sub

Sub Main
    Call SayHello
End Sub
```

The yellow execution arrow shows where execution has stopped. Click the Run Under Debugger button to continue execution. The script will stop on a breakpoint within the SayHello sub in script B.

```
Sub SayHello
    str = "Hello!"
    Application.MessageBox str
End Sub

Sub Main
    Call SayHello
End Sub
```

10. Let’s take a look at the Call Stack pane. The first line reads “SayHello” while both the second and third line read “Main”. Press in the Show Module button to show the module (script) name for each call. Press in the Show Line Position button to show the position of a call within a line (useful when you have more than one call on the same line). Click the Show Language button to show the language of each script module. Click on the last call, for module A function Main line 4, to highlight it and click the Go To button in the Call Stack pane toolbar. This will activate script A and mark the location of this call in the script window with a green arrow:

```
Sub Main
    Set components = Application.ActiveDocument.ComponentSet
    Set anotherScript = components(components.ItemByName("B"))
    anotherScript.Run
End Sub
```

It will also mark the respective item in the Call Stack pane with a green arrow as well:
The Variables pane will display "components" and "anotherScript" declared within the scope of this call. The "components.Count+1" entry in the Watches pane will read "3". Double-click the second line within the Call Stack pane, reading module B, function Main, line 7. Double-clicking a line in the Call Stack pane is an alternative way to Go To a call. This will activate script B.

Note that line 7 that contains the current call is marked with a green arrow and line 3 containing the current execution line is still displayed with a yellow arrow overlaid on the breakpoint dot. The Variables pane will display no entries since the Main routine in script B declares no local variables. The components.Count + 1 entry within the Watches pane will read "Variable is undefined: components" since "components" is inaccessible within the scope of the Main routine in script B.

Double-click the first line within the Call Stack pane, for module B, function SayHello and line 3. The Variables pane will display the name str and value "Hello!" as well as the type String if the Type column is on. The Watches pane will still refuse to evaluate the value of the components.Count+1 expression, since it inaccessible with the scope of the SayHello routine.

11. In the Watches pane add a watch for the str variable. This will display "Hello!". Now, double-click the value box of the "str" variable (that is, click on the cell with "Hello!" in it) either within the Variables pane or within the Watches pane. Change "Hello!" to just "Hi!" and press Enter. Both Variables and Watches will now display the value of str as "Hi!". Click the Run Under Debugger button in the Tools toolbar to continue execution. Note that greeting in the message box has also changed to "Hi!". Click OK to dismiss the message and finish the script.

12. Note that all watches will turn to "Unavailable". We can still add or remove watches but we can not (obviously) edit their content. The Call Stack and Variables panes will disable their lists.

**Installation Requirements**

The Manifold Debugger is designed to work with Microsoft ActiveX scripting languages as used within Manifold System and to take advantage of the debugging capabilities Microsoft has built into the ActiveX ensemble of technologies.

To run the Debugger, you must therefore have the Microsoft Process Debug Manager (PDM) DLL installed on your system. This DLL gets installed with various Microsoft products aimed at scripting development. If you don't have the PDM or don't know if you have it, you must install the PDM on your machine. Unfortunately, due to what appears to be an inadvertent error in Microsoft licensing, although the PDM is freely usable and obtainable from Microsoft it is not included in the "redistributables" list covering all other Microsoft ActiveX components issued to developers like manifold.net for redistribution to end users. manifold.net therefore cannot provide the PDM as part of the Debugger installation on CD.
The easiest way to assure you have the PDM correctly installed on your system is to download any of the free Microsoft items that install it. The recommended choice is to install the Microsoft Script Debugger for the Windows system you are using from the following URL (making sure to enter the entire URL as a single line in your browser):


The above URL was correct at the time of this writing. If it is no longer active, visit the Microsoft web site for a current URL, or search for "Microsoft Script Debugger" using Google or some other search engine to find Microsoft's new download page. Note that the debugger comes in different versions for different Microsoft operating systems, so make sure you get the version that is correct for your system. For example, if running Windows XP make sure you get the version for Windows XP and not the one for Windows 98 or other Windows edition.

Note that at the present writing the Microsoft Script Debugger is not available for either 64-bit Windows systems or Vista.

You must have Administrator privileges in Windows 2003, XP and 2000 when installing the script debugger.

Note: The Microsoft Script Debugger is a small, simple, command-line debugger that also uses the PDM. It is more primitive than the Manifold Debugger, but it does install the PDM DLL required by the Manifold Debugger. Users new to working with debuggers may also find the documentation links available from the URL above to be a helpful introduction to debugging concepts used in the Microsoft scripting world by the Manifold Debugger and all other standard ActiveX scripting tools.

See Also

Panes used with the debugger include:

Call Stack
Variables
Watches

Extensions

Extensions

Manifold can be extended in the following ways:

- By writing add-ins. Add-ins are short scripts that users can run in the context of the currently opened document. Add-ins can be written using any of the scripting languages supported by Manifold, including C#, VB .NET, VBScript, JScript and others, and can utilize external libraries and objects, via .NET, COM and other technologies.

- By writing geocoding server modules. Geocoding server modules are .NET modules with objects supporting the Manifold Geocoding Server Interface. Geocoding server modules are used to connect to external geocoding services for geocoding address data.

- By writing image server modules. Image server modules are .NET modules with objects supporting the Manifold Image Server Interface. Image server modules are used to connect to image data sources and expose their data as image components.

While add-ins are only available for use within interactive Manifold sessions, and can not be used as they are from external applications utilizing the Manifold object model or from IMS web sites, geocoding server modules and image server modules are available for use both in interactive Manifold sessions as well as programmatically from external applications and web sites.

See Also
Add-Ins
Geocoding Server Interface
Image Server Interface
Geocoding Server Interface

The Manifold Geocoding Server interface is a .NET interface used with geocoding server modules. Geocoding server modules connect to external geocoding services and geocode address data. For example, geocoding server modules can act as intermediaries between Manifold and various web sites that expose programmatic interfaces for doing simple geocoding tasks.

A Manifold Geocoding Server module should include one or more .NET objects exposing the Manifold Geocoding Server interface. Each of the objects should have a public parameterless constructor.

Installing a Manifold geocoding server module consists of installing the .NET assembly that contains .NET objects exposing the Manifold Geocoding Server interface into the Manifold installation folder, and doing the necessary configuration required by the assembly. There is no configuration required on the Manifold side besides copying the assembly into the Manifold installation folder and restarting Manifold.

Connecting to a Manifold geocoding server consists of creating an instance of the particular .NET object, and calling its methods to geocode data.

Interface Specification

The Manifold Geocoding Server interface includes the following members:

IServer

IResults Geocode(String address)

Geocode the specified address and return a set of matches.

String Country

Get default country. Return a blank string if the server supports more than one country.

String Name

Get the unique name of the geocoding server module. Only alphanumeric characters, spaces, and ., -, :, _, <, >, '{', '}', '[' and ']' characters are permitted.

Boolean Remote

Return True if Manifold should minimize the number of geocoding requests, and False otherwise. Commonly used with servers accessed via the web or via slow network connections.

(Supplementary interfaces)

IResult

String Address

Get street address.

String City

Get city name.

String Country

Get country name.

Double Latitude

Get latitude value from -90 to 90. If latitude value is unknown, return InvalidCoord.

Double Longitude

Get longitude value from -180 to 180. If longitude value is unknown, return InvalidCoord.
Programming

String State
Get state name or province name.

String Zip
Get zip code or postal code.

IResults

Int32 Count
Get number of available results.

IResult this[Int32 index]
Get result with given index, from 0 to Count-1. This is an indexer property.

The definition of the main Geocoding Server interface and supplementary interfaces resides in Manifold.GeocodingServer.dll.

All interfaces are defined in the Manifold.GeocodingServer namespace.

Notes

Implement all members of the Manifold Geocoding Server interface, or the geocoding server module will not be functional. Pay close attention to the meaning of each property and method.

Make sure to name your geocoding server module in a unique way. A common technique is to use the name of your company or project as part of the name for a geocoding server module, for example, Acme Corp Geocoding Server for Canada. Do not use any characters that are not explicitly permitted.

Limit the amount of memory used by your geocoding server module to a minimum. If you want to cache the results returned by the server, be sure to design and follow a robust caching policy. Do not blindly cache every result without ever pruning the cache.

Limit the amount of bandwidth used by your geocoding server module. Return True in the Remote property if you are going to use a web connection or a slow network connection.

Design your geocoding server module to operate with minimal permissions. At the very least, your module should operate in the context of a normal user account. It is frequently beneficial to declare all permissions required by the module via .NET mechanisms, for use in limited-permission environments.

Design your geocoding server module so it does not require any configuration. If you want the user to be able to customize the operation of the module, provide an interactive configuration tool. Consider storing configuration options in isolated storage, using per-user isolation.

Programmers should visit the manifold.net website for any supplemental documentation or example implementations, if any are made available in the future.

See Also

Geocoding Data Sources
Extensions
Image Server Interface
Manifold Geocoding Servers
Street Address Geocoding
The Manifold Image Server interface is a .NET interface used with image server modules. Image server modules connect to image data sources and expose their data as image components that can be used together with other components inside Manifold projects. For example, there are image server modules for bringing image data into Manifold from various online mapping services such as Google Maps, Yahoo! Maps or Microsoft Virtual Earth.

A Manifold image server module should include one or more .NET objects exposing the Manifold Image Server interface. Each of the objects should have a public parameterless constructor.

Installing a Manifold image server module consists of installing the .NET assembly that contains .NET objects exposing the Manifold Image Server interface into the Manifold installation folder, and doing the necessary configuration required by the assembly. There is no configuration required on the Manifold side besides copying the assembly into the Manifold installation folder and restarting Manifold.

Connecting to a Manifold image server consists of creating an instance of the particular .NET object, setting its properties, retrieving the dimensions and supplementary information describing the target image, and requesting pixels of the target image by tiles.

An image server can choose to operate at one or more scale levels. The resolution of a particular scale level should be 2 times that of the next available scale level. All tiles have the same width and height, regardless of the scale level. Tile coordinates are required to be non-negative.

Image tiles can be in any of the following formats: BMP, ECW, GIF, J2K (JPEG 2000), JPG, PNG, TGA, TIF.

Images linked via the image server interface will use whatever coordinate system (projection) is returned by the image server module in use. If an invalid coordinate system is returned when linking an image Manifold will use the default coordinate system.

**Interface Specification**

The Manifold Image Server interface includes the following members:

**IServer**

String CoordinateSystem
Get the coordinate system of the image in Manifold XML format.

String DefaultImageType
Get default image type, such as ".jpg" or ".png".

String DefaultURL
Get default URL, such as "http://www.example.com/myserver.asp".

Boolean DownloadTile(Int32 x, Int32 y, Int32 scale, String filename)
Download the specified image tile and save it at the specified location in the file system.

String Error
Get last error message.

IRectangle GetRectPixels(Int32 scale, IRectangleD rect)
Convert the supplied rectangle from the coordinate system of the image to pixels. The XMax property of the returned rectangle should be equal to or greater than the XMin property, same for YMax and YMin. Returning a rectangle of (XMin=500, YMin=500, XMax=599, YMax=549) means that the queried image area is 100 pixels wide and 50 pixels high, at the supplied scale level.

IRectangle GetRectTiles(Int32 scale, IRectangleD rect)
Manifold® System Release 8.00 User Manual

Convert the supplied rectangle from the coordinate system of the image to tiles. The XMax property
of the returned rectangle should be equal to or greater than the XMin property, same for YMax and
YMin. Returning a rectangle of (XMin=5, YMin=5, XMax=7, YMax=5) means that the queried image
area is 3 tiles wide and 1 tile high, at the supplied scale level.
String Name
Get the unique name of the image server module. Only alphanumeric characters, spaces, and ".", "", ":", "_", "<", ">", "(", ")", "[", "]", "{", "}", "?", "!", "~", "@", "#", "$", "%", "^", "&", "|" characters are
permitted.
String ProxyAddress
Get or set the address of the proxy server.
String ProxyPassword
Get or set the password for use with the proxy server.
String ProxyUserName
Get or set the user name for use with the proxy server.
Boolean ReverseY
Return True if the pixels and tiles with smaller values of the Y coordinate are on top of those with
larger values of the Y coordinate (the Y axis points down), and False otherwise.
Int32 ScaleHi
Get the least detailed scale. Should return a value equal to or larger than ScaleLo.
Int32 ScaleLo
Get the most detailed scale. Should return a value equal to or less than ScaleHi.
String ScaleNames
Get names of scales, from most detailed to least detailed, separated by commas, for example, "1
meter,2 meters".
Int32 TileSizeX
Get the width of a single tile, in pixels.
Int32 TileSizeY
Get the height of a single tile, in pixels.
String URL
Get or set the URL for the target image.
(Supplementary interfaces)
IRectangle
Int32 XMax
Get maximum X coordinate.
Int32 XMin
Get minimum X coordinate.
Int32 YMax
Get maximum Y coordinate.

2516


Int32 YMin
Get minimum Y coordinate.

IRectangleD

Double XMax
Get maximum X coordinate.

Double XMin
Get minimum X coordinate.

Double YMax
Get maximum Y coordinate.

Double YMin
Get minimum Y coordinate.

The definition of the main Image Server interface and supplementary interfaces resides in Manifold.ImageServer.dll.

All interfaces are defined in the Manifold.ImageServer namespace.

Notes

Implement all members of the Manifold Image Server interface, or the image server module will not be functional. Pay close attention to the meaning of each property and method.

Make sure to name your image server module in a unique way. A common technique is to use the name of your company or project as part of the name for an image server module, for example, Acme Corp Web Map. Do not use any characters that are not explicitly permitted.

Limit the amount of memory used by your image server module to a minimum. There is no need to cache image data, since Manifold already implements caching.

Play well with proxy servers. Do not ignore the values of the ProxyXxx properties supplied by Manifold.

Design your image server module to operate with minimal permissions. At the very least, your module should operate in the context of a normal user account. It is frequently beneficial to declare the permissions required by the module via .NET mechanisms, for use in limited-permission environments.

Design your image server module so it does not require any configuration. If you want the user to be able to customize the operation of the module, provide an interactive configuration tool. Consider storing configuration options in isolated storage, using per-user isolation.

Programmers should also visit the manifold.net website for any supplemental documentation or example implementations, if any are made available in the future.

See Also

Extensions
Geocoding Server Interface
Linked Images from Manifold Image Servers
Programming Reference

For general information on programming Manifold, see the Programming Manifold topic as well as the Scripts and Forms topics.

The Scripting Reference topic provides a reference list of constants, objects and controls information.
Scripting Reference

The topics for many objects will include an Obtained From section at the end of the topic that is a reverse index showing how to obtain that object.

A fairly frequent situation is to know we would like to use a method but we might not know how to get to the object in which that method is implemented. For example we might know that we need to call Intersect but we do not know what to call to use to obtain a GeomSet object which implements the Intersect method. The Obtained From section provides a guide to the objects and methods used to create or otherwise obtain a GeomSet object.

All COM objects exposed by Manifold use apartment-model threading.

Notes

All .NET assemblies (including Manifold.Interop) are installed into the global assembly cache to facilitate reuse.

A .NET script invoked in the context of a web site can access the Document property of the Context object.

A .NET script can access the running script component using the Script property of the Context object. If the script is an add-in script, both the Script and the Document properties of the Context object return null values.

A non-.NET script can access the global Script object, which represents the running script, and the global Document object, which represents the document containing the running script. If the script is an add-in script, both the Script and the Document objects are empty.

New releases of Manifold may introduce new formatting styles. The /alist command line option provides a handy way for programmers to get a comprehensive list of all area, label, line and point styles available and their names.
Constants

Border Constants

Printing borders.

Values

BorderDouble

Double line border.

BorderNone

No border.

BorderSingle

Single line border.

BorderSingleFrame

Single line border with padding.

See Also

Scripting Reference
**BufferType Constants**
Types of buffers used in geometric computations.

**Values**

BufferTypeBorder
  Buffer around border.

BufferTypeInner
  Inner buffer.

BufferTypeOuter
  Outer buffer.

**See Also**

Scripting Reference
ByteOrder Constants
Byte orders used in converters.

Values

LSB
Intel byte order (least significant byte first).

MSB
Motorola byte order (most significant byte first).

See Also

Scripting Reference
CameraType Constants
Types of cameras used in terrains.

Values

CameraTypeFly
Flying camera (no vertical bound).

CameraTypeWalk
Walking camera (bounded to surface).

See Also

Scripting Reference
ChannelInterleaving Constants
Channel interleaving options used in image converters (multichannel imports only).

Values

BIL
Line interleaving. Multiple channels are combined at line level.

BIP
Pixel interleaving. Multiple channels are combined at pixel level.

BSQ
Channel interleaving. Multiple channels are stored after each other.

See Also

Scripting Reference
**ColumnAlign Constants**
Alignment options used in table columns.

**Values**

- **ColumnAlignCenter**
  - Column text is centered.

- **ColumnAlignLeft**
  - Column text is aligned to left.

- **ColumnAlignRight**
  - Column text is aligned to right.

**See Also**

Scripting Reference
**ColumnCategory Constants**
Table column categories.

**Values**

**ColumnCategoryDss**
Column is computed by DSS engine (rank column).

**ColumnCategoryForeign**
Column is mapped from another table.

**ColumnCategoryIntrinsic**
Column is an intrinsic property of respective object.

**ColumnCategoryLink**
Column is linked from the external OLEDB data source.

**ColumnCategoryMask**
Column stores record mask.

**ColumnCategoryNative**
Column is native.

**ColumnCategoryScript**
Column is computed by the script (active column).

**ColumnCategorySql**
Column is computed by the query engine.

**See Also**

Scripting Reference
ColumnFlag Constants
Flags used in types of table columns. For possible column types see ColumnType Constants.

Values

ColumnFlagAChar
ANSI character.

ColumnFlagBoolean
Boolean type modifier.

ColumnFlagCurrency
Currency.

ColumnFlagFloat32
Single precision floating-point number.

ColumnFlagFloat64
Double precision floating-point number.

ColumnFlagInt16
16 bit integer number.

ColumnFlagInt32
32 bit integer number.

ColumnFlagInt8
8 bit integer number.

ColumnFlagLatitude
Latitude type modifier.

ColumnFlagLongitude
Longitude type modifier.

ColumnFlagPercent
Percentage type modifier.

ColumnFlagTime
Date and time.

ColumnFlagUnsigned
Unsigned type modifier.

ColumnFlagUrl
URL type modifier.

ColumnFlagVector
Vector type modifier.

ColumnFlagWChar
Unicode character.
See Also

Scripting Reference
**ColumnFormatNeg Constants**
Formatting options for negative values.

**Values**

- FormatNegBracketBuckSpaceValueBracket
  $(2.345)$

- FormatNegBracketBuckValueBracket
  $(2.345)$

- FormatNegBracketValueBracket
  (2.345)

- FormatNegBracketValueBuckBracket
  (2.345$)

- FormatNegBracketValueSpaceBuckBracket
  (2.345 $)

- FormatNegBuckMinusValue
  $-2.345$

- FormatNegBuckSpaceMinusValue
  $-2.345$

- FormatNegBuckSpaceValueMinus
  $2.345-$

- FormatNegBuckValueMinus
  $2.345-$

- FormatNegDefault
  Default formatting set in Regional Options.

- FormatNegMinusBuckSpaceValue
  -$2.345$

- FormatNegMinusBuckValue
  -$2.345$

- FormatNegMinusSpaceValue
  -2.345

- FormatNegMinusValue
  -2.345

- FormatNegMinusValueBuck
  -2.345$

- FormatNegMinusValueSpaceBuck
  -2.345 $
FormatNegValueBuckMinus
2.345$-

FormatNegValueMinus
2.345-

FormatNegValueMinusBuck
2.345$-

FormatNegValueMinusSpaceBuck
2.345- $

FormatNegValueSpaceBuckMinus
2.345 $-

FormatNegValueSpaceMinus
2.345 -

See Also

Scripting Reference
ColumnFormatPos Constants
Formatting options for positive values.

Values

FormatPosBuckSpaceValue
$ 2.345

FormatPosBuckValue
$2.345

FormatPosDefault
Default formatting set in Regional Options.

FormatPosValueBuck
2.345$

FormatPosValueSpaceBuck
2.345 $
**ColumnType Constants**

Types of table columns. Each type is a combination of one or more of the ColumnFlag Constants.

**Values**

ColumnTypeAChar
ANSI character.

ColumnTypeAText
ANSI text (variable-length or fixed-length).

ColumnTypeBinary
Binary data (variable-length or fixed-length).

ColumnTypeBoolean
Boolean value.

ColumnTypeCoordSys
Coordinate system.

ColumnTypeCurrency
Currency.

ColumnTypeFloat32
Single precision floating-point number.

ColumnTypeFloat64
Double precision floating-point number.

ColumnTypeGeom
Geometric shape.

ColumnTypeGeomSDE
Geometric shape in ESRI SDE format.

ColumnTypeGeomSHP
Geometric shape in ESRI SHP format.

ColumnTypeGeomWKB
Geometric shape in OpenGIS WKB format.

ColumnTypeInt16
16 bit integer number.

ColumnTypeInt16U
Unsigned 16 bit integer number.

ColumnTypeInt32
32 bit integer number.

ColumnTypeInt32U
Unsigned 32 bit integer number.
ColumnTypeInt8
  8 bit integer number.

ColumnTypeInt8U
  Unsigned 8 bit integer number.

ColumnTypeLatitude
  Latitude value.

ColumnTypeLongitude
  Longitude value.

ColumnTypePercentage
  Percentage value.

ColumnTypeTime
  Date and time value.

ColumnTypeURL
  URL string.

ColumnTypeWChar
  Unicode character.

ColumnTypeWText
  Unicode string.

See Also

Scripting Reference


**ComponentType Constants**
Types of components within the map file.

**Values**

- ComponentChart
  Chart.

- ComponentComments
  Comments.

- ComponentDrawing
  Drawing.

- ComponentElevation
  Elevation.

- ComponentFolder
  Folder.

- ComponentForm
  Form.

- ComponentImage
  Image.

- ComponentLabels
  Labels.

- ComponentLayout
  Layout.

- ComponentMap
  Map.

- ComponentNull
  Invalid value. Typically used in initialization.

- ComponentPalette
  Palette.

- ComponentProfile
  Profile.

- ComponentQuery
  Query.

- ComponentScript
  Script.

- ComponentSurface
  Surface.
ComponentTable
   Table.

ComponentTerrain
   Terrain.

ComponentTheme
   Theme.

ComponentZones
   Zones.

See Also

Scripting Reference
**ComputationMode Constants**

Computation mode of an active column.

**Values**

- **ComputationModeCachedUpdateable**
  Computes active column on demand. Caches computed values and resets cache whenever script component is changed.

- **ComputationModeCached**
  Computes active column on demand. Caches computed values.

- **ComputationModeManual**
  Computes active column on user request.

- **ComputationModeUncached**
  Computes active column on demand. Does not cache values.

**See Also**

Scripting Reference
**ConvertPolicy Constants**
Import or export policies used in converters.

**Values**

ConvertAll
Import or export all items affected by policy.

ConvertNone
Import or export none of items affected by policy.

ConvertSome
Import or export explicit set of items supplied with ConverterItemSet.

**See Also**

Scripting Reference
**ConvertPrompt Constants**
Prompt options used in converters.

**Values**

- **PromptAlways**
  Always display conversion dialog.

- **PromptDefault**
  Display conversion dialog according to options.

- **PromptNone**
  Never display conversion dialog.

**See Also**

Scripting Reference
CoordinateBand Constants
Coordinate band types used in printing.

Values

CoordinateBandLatLon
Print latitude/longitude coordinates.

CoordinateBandMetric
Print native coordinates.

CoordinateBandNone
Do not print any coordinates.

See Also

Scripting Reference
CoordinateSystemParameterType Constants
Types of coordinate system parameters.

Values

ParameterLatitude
Latitude. Parameter value is in degrees.

ParameterLength
Length. Parameter value is in meters.

ParameterLongitude
Longitude. Parameter value is in degrees.

ParameterOther
Other.

See Also

Scripting Reference
DSSAtomType Constants
Types of atomic expressions used in rank columns.

Values

DSSAtomTypeAverage
Ranks average values higher than smaller or greater ones.

DSSAtomTypeCurve
Ranks values according to user-supplied curve.

DSSAtomTypeHigh
Ranks greater values higher than smaller values.

DSSAtomTypeLow
Ranks smaller values higher than greater values.

DSSAtomTypeTwoPoints
Ranks values according to a curve defined by two points.

See Also

Scripting Reference
**DSSHedge Constants**
Hedges applied to criteria in rank columns.

**Values**

DSSHedgeAbove
Ranks greater values higher than smaller values.

DSSHedgeBelow
Ranks smaller values higher than greater values.

DSSHedgeExtremely
Greatly increases selectiveness of subject criterion.

DSSHedgeNone
Has no effect on subject criterion.

DSSHedgeSlightly
Decreases selectiveness of subject criterion.

DSSHedgeSomewhat
Greatly decreases selectiveness of subject criterion.

DSSHedgeVery
Increases selectiveness of subject criterion.

**See Also**

Scripting Reference
**DSSJunction Constants**
Junctions applied to criteria in rank columns.

**Values**

DSSJunctionAnd
AND junction.

DSSJunctionNone
No junction (used for the last criterion).

DSSJunctionOr
OR junction.

**See Also**

Scripting Reference
FormatType Constants
Type of formatting.

Values

FormatTypeNone
Unknown / uninitialized.

FormatTypeColor
Color.

FormatTypeAreaSize
Area size.

FormatTypeAreaStyle
Area style.

FormatTypeLabelRotation
Label rotation angle.

FormatTypeLabelSize
Label size.

FormatTypeLabelStyle
Label style.

FormatTypeLineSize
Line size.

FormatTypeLineStyle
Line style.

FormatTypePointRotation
Point rotation angle.

FormatTypePointSize
Point size.

FormatTypePointStyle
Point style.

See Also

Scripting Reference
GeocodeLevel Constants
Geocoding levels.

Values

GeocodeLevelBuilding
Location is a building.

GeocodeLevelCity
Location is a city (place) centroid.

GeocodeLevelNoLocation
Location is not available.

GeocodeLevelStreet
Location is a street centroid.

GeocodeLevelZip
Location is a zip centroid.

See Also

Scripting Reference
GeocodeStatus Constants
Results of a geocoding operation.

Values

GeocodeCriticalError
Critical error.

GeocodeSuccess
Success.

GeocodeUnknownBuilding
Unknown building.

GeocodeUnknownStreet
Unknown street name, no similar names.

GeocodeUnknownStreetMisspelling
Unknown street name, possible misspelling.

GeocodeUnknownZip
Unknown zip code.

See Also

Scripting Reference
**GeomArea, GeomLine and GeomPoint Constants**
Types of geometric entities.

**Values**

GeomArea
Area (polygon).

GeomLine
Line (polyline).

GeomPoint
Point.

**See Also**

Scripting Reference
**ImageType Constants**
Types of image components.

**Values**

ImageTypeGrayscale
Grayscale image.

ImageTypePalette
Palette image.

ImageTypeRGB
Color image.

ImageTypeRGBA
Color image with alpha channel.

**See Also**

Scripting Reference
JpegCompression Constants
JPEG compression methods used in converters.

Values

JpegCompressionBaseline
Baseline compression.

JpegCompressionBaselineOptimized
Optimized baseline compression.

JpegCompressionProgressive
Progressive scans.

See Also

Scripting Reference
**JpegQuality Constants**

JPEG quality values used in converters.

**Values**

JpegQualityHigh
High quality.

JpegQualityLow
Low quality.

JpegQualityMax
Maximum possible quality.

JpegQualityMed
Medium quality.

**See Also**

Scripting Reference
LabelAlignMulti Constants
Alignment options for multiline labels.

Values

LabelAlignMultiCenter
Multiple lines of text are centered.

LabelAlignMultiJustify
Multiple lines of text are justified with whitespace.

LabelAlignMultiLeft
Multiple lines of text are aligned to left side.

LabelAlignMultiRight
Multiple lines of text are aligned to right side.

See Also

Scripting Reference
**LabelAlignX Constants**
Horizontal alignment options for labels.

**Values**

LabelAlignXCenter
Label is centered over the tie point.

LabelAlignXLeft
Label is at the left of tie point.

LabelAlignXRight
Label is at the right of tie point.

**See Also**

Scripting Reference
LabelAlignY Constants
Vertical alignment options for labels.

Values

LabelAlignYBottom
Label is at the bottom of tie point.

LabelAlignYCenter
Label is centered vertically.

LabelAlignYTop
Label is at the top of tie point.

See Also

Scripting Reference
LayoutBorder Constants
Layout borders.

Values

LayoutBorderCoordinates
Coordinates given by the user.

LayoutBorderCoordinatesGraticule
Coordinates taken from graticule.

LayoutBorderCoordinatesGrid
Coordinates taken from grid.

LayoutBorderNone
No border.

LayoutBorderThick
Thick (1 point) border.

LayoutBorderThin
Thin (1/20 point) border.

See Also

Scripting Reference
**LayoutPaging Constants**

Layout paging modes.

**Values**

- **LayoutPagingContinuous**
  - Continuous paging mode.

- **LayoutPagingIndividual**
  - Individual paging mode.

**See Also**

Scripting Reference
LayoutScope Constants
Layout scopes.

Values

LayoutScopeAll
All objects or pixels.

LayoutScopeBox
Fixed area whose rectangle is specified by the ScopeArea property. You can also read or write the center / scale of the area using the ScopeCenter and ScopeScale properties.

LayoutScopeLayer
Layer specified by the ScopeDetail property.

LayoutScopeScale
Fixed area whose center / scale is specified by the ScopeCenter and ScopeScale properties. You can also read or write the area using the ScopeArea property.

LayoutScopeSelection
Selection.

LayoutScopeSavedSelection
Saved selection specified by the ScopeDetail property.

LayoutScopeView
View specified by the ScopeDetail property.

See Also

Scripting Reference
**LayoutState Constants**
State of a layout entry option like graticule or legend.

**Values**

- **LayoutStateCustom**
  Option is overridden by the layout entry (applies to background).

- **LayoutStateOff**
  Option is turned off.

- **LayoutStateOn**
  Option is turned on.

- **LayoutStateOnWithComponent**
  Option is controlled by the component ("auto").

**See Also**

Scripting Reference
**LayoutType Constants**
Type of layout entry.

**Values**

LayoutTypeBody
Owner component.

LayoutTypeComponent
Arbitrary component.

LayoutTypeLegend
Legend (should be bound to some other layout entry).

LayoutTypeLineHorz
Horizontal line.

LayoutTypeLineVert
Vertical line.

LayoutTypeNorthArrow
North arrow (should be bound to some other layout entry).

LayoutTypeNull
Invalid type.

LayoutTypeRect
Rectangle.

LayoutTypeScaleBar
Scale bar (should be bound to some other layout entry).

LayoutTypeText
Text (can be bound to some other layout entry).

**See Also**

Scripting Reference
LegendType Constants
Legend types.

Values

LegendStandard
Standard legend.

See Also

Scripting Reference
LightingDirection Constants
Lighting directions used in surfaces and terrains.

Values

DirectionEast
Sun shines from east.

DirectionNone
Invalid value. Typically used in initialization.

DirectionNorth
Sun shines from north.

DirectionNorthEast
Sun shines from north-east.

DirectionNorthWest
Sun shines from north-west.

DirectionSouth
Sun shines from south.

DirectionSouthEast
Sun shines from south-east.

DirectionSouthWest
Sun shines from south-west.

DirectionWest
Sun shines from west.

See Also

Scripting Reference
LinkEditMode Constants
Editing modes used by linked components.

Values

LinkEditModeNone
Invalid editing mode.

LinkEditModeOverwrite
Editing the component overwrites changes made by others.

LinkEditModeReadOnly
The component is read-only. No editing is possible.

LinkEditModeReview
Editing the component detects changes made by others and resolves editing conflicts using the Review pane.

See Also

Scripting Reference
MapServerRenderFormat Constants
Rendering format used by the MapServer object.

Values

MapServerRenderFormatDefault
Default render format (PNG).

MapServerRenderFormatGif
GIF render format.

MapServerRenderFormatJpeg
JPEG render format, supports specifying rendering quality.

MapServerRenderFormatPng
PNG render format.

See Also

Scripting Reference
MessageBoxResult Constants
Status codes returned by the MessageBox and MessageBoxEx methods of the Application object.

Values

MessageBoxResultAbort
Abort button has been pressed.

MessageBoxResultCancel
Cancel button has been pressed.

MessageBoxResultIgnore
Ignore button has been pressed.

MessageBoxResultNo
No button has been pressed.

MessageBoxResultOk
OK button has been pressed.

MessageBoxResultRetry
Retry button has been pressed.

MessageBoxResultYes
Yes button has been pressed.

See Also

Scripting Reference
MessageBoxType Constants
Icons and buttons used by message boxes invoked by the MessageBox and MessageBoxEx methods of the Application object. Individual constants can be combined using the binary "or" operator, or, in case of combining a button or a set of buttons and an icon, using the ordinary addition operator.

Values

MessageBoxTypeAbortRetryIgnore
Abort, Retry and Ignore buttons.

MessageBoxTypeApplModal
The user has to close the message box before continuing to work with the application.

MessageBoxTypeCancelTryContinue
Cancel, Try and Continue buttons.

MessageBoxTypeDefaultButton1
Make first button the default button.

MessageBoxTypeDefaultButton2
Make second button the default button.

MessageBoxTypeDefaultButton3
Make third button the default button.

MessageBoxTypeDefaultButton4
Make fourth button the default button.

MessageBoxTypeDefaultDesktopOnly
Display the message box only on the default desktop.

MessageBoxTypeHelp
Help button.

MessageBoxTypeIconAsterisk
Information icon.

MessageBoxTypeIconError
Stop icon.

MessageBoxTypeIconExclamation
Exclamation icon.

MessageBoxTypeIconHand
Stop icon.

MessageBoxTypeIconInformation
Information icon.

MessageBoxTypeIconQuestion
Question icon.

MessageBoxTypeIconStop
Stop icon.

MessageBoxTypeIconWarning
   Exclamation icon.

MessageBoxTypeNoFocus
   Display the message box without making it focused.

MessageBoxTypeOk
   OK button.

MessageBoxTypeOkCancel
   OK and Cancel buttons.

MessageBoxTypeRetryCancel
   Retry and Cancel buttons.

MessageBoxTypeRight
   Right-justify message box text.

MessageBoxTypeRTLReading
   Force right-to-left reading order.

MessageBoxTypeSetForeground
   Bring the message box window to the foreground.

MessageBoxTypeSystemModal
   The user has to close the message box before continuing to work with the application. Makes the message box window topmost.

MessageBoxTypeTaskModal
   The user has to close the message box before continuing to work with the application. Disables all top-level windows belonging to the thread that launched the message box.

MessageBoxTypeTopmost
   Make the message box window topmost.

MessageBoxTypeYesNo
   Yes and No buttons.

MessageBoxTypeYesNoCancel
   Yes, No and Cancel buttons.

See Also

Scripting Reference
NorthArrowType Constants

North arrow types.

Values

NorthArrowCompass
Compass pointing North, South, East and West.

NorthArrowCompassN
Compass pointing North, South, East and West with a letter N.

NorthArrowCompassNS
Compass pointing North, South, East and West with letters N and S.

NorthArrowCompassNESW
Compass pointing North, South, East and West with letters N, S, E and W.

NorthArrowDouble
Arrow pointing North and South.

NorthArrowDoubleN
Arrow pointing North and South with a letter N.

NorthArrowDoubleNS
Arrow pointing North and South with letters N and S.

NorthArrowDoubleShadow
Arrow pointing North and South with a shadow.

NorthArrowDoubleShadowN
Arrow pointing North and South with a letter N and a shadow.

NorthArrowDoubleShadowNS
Arrow pointing North and South with letters N and S and a shadow.

NorthArrowHalfHarpoon
Simple "half-harpoon" arrow pointing North.

NorthArrowHalfHarpoonN
Simple "half-harpoon" arrow pointing North with a letter N.

NorthArrowHalfHarpoonNS
Simple "half-harpoon" arrow pointing North with letters N and S.

NorthArrowHarpoon
Simple "harpoon" arrow pointing North.

NorthArrowHarpoonN
Simple "harpoon" arrow pointing North with a letter N.

NorthArrowHarpoonNS
Simple "harpoon" arrow pointing North with letters N and S.
NorthArrowNaval
Naval arrow pointing North and South.

NorthArrowNavalN
Naval arrow pointing North and South with a letter N.

NorthArrowNavalNS
Naval arrow pointing North and South with letters N and S.

NorthArrowRose
Rose compass pointing North, South, East and West.

NorthArrowRoseArrow
North arrow from rose compass.

NorthArrowRoseArrowN
North arrow from rose compass with a letter N.

NorthArrowRoseN
Rose compass pointing North, South, East and West with a letter N.

NorthArrowRoseNS
Rose compass pointing North, South, East and West with letters N and S.

NorthArrowRoseNESW
Rose compass pointing North, South, East and West with letters N, S, E and W.

NorthArrowRoseQuad
Rose compass without intermediate junctions pointing North, South, East and West.

NorthArrowRoseQuadN
Rose compass without intermediate junctions pointing North, South, East and West with a letter N.

NorthArrowRoseQuadNS
Rose compass without intermediate junctions pointing North, South, East and West with letters N and S.

NorthArrowRoseQuadNESW
Rose compass without intermediate junctions pointing North, South, East and West with letters N, S, E and W.

NorthArrowRoseTwin
North and South arrows from rose compass without intermediate junctions.

NorthArrowRoseTwinN
North and South arrows from rose compass without intermediate junctions with a letter N.

NorthArrowRoseTwinNS
North and South arrows from rose compass without intermediate junctions with letters N and S.

NorthArrowSingle
Arrow pointing North.

NorthArrowSingleN
Arrow pointing North with a letter N.

NorthArrowSingleShadow
Arrow pointing North with a shadow.

NorthArrowSingleShadowN
Arrow pointing North with a letter N and a shadow;

See Also

Scripting Reference
**Object Type Constants**
Types of drawing objects.

**Values**

ObjectArea
Area.

ObjectLine
Line.

ObjectPoint
Point.

**See Also**

Scripting Reference
PageIndicator Constants
Types of page indicator used in printing.

Values

PageIndicatorGraphic
Graphic page indicator.

PageIndicatorNone
No page indicator.

PageIndicatorText
Textual page indicator (eg. "A1").

See Also

Scripting Reference
PointUSNGPrecision Constants
USNG precision levels.

Values

PointUSNGPrecision10km
10 kilometers precision.

PointUSNGPrecision1km
1 kilometer precision.

PointUSNGPrecision100m
100 meters precision.

PointUSNGPrecision10m
10 meters precision.

PointUSNGPrecision1m
1 meter precision.

See Also

Scripting Reference
PsPixelEncoding Constants
Pixel encoding used in PS files.

Values

PsEncodingAsciiHex
ASCII Hex.

PsEncodingFlat
Flat.

PsEncodingRunLength
Run-length.

See Also

Scripting Reference
**QueryType Constants**
Query types.

**Values**

- **QueryTypeAlterTable**
  ALTER TABLE query.

- **QueryTypeCreateDrawing**
  CREATE DRAWING query.

- **QueryTypeCreateView**
  CREATE VIEW query.

- **QueryTypeCreateTable**
  CREATE TABLE query.

- **QueryTypeCrossTable**
  TRANSFORM query.

- **QueryTypeDelete**
  DELETE query.

- **QueryTypeDropTable**
  DROP TABLE query.

- **QueryTypeDropView**
  DROP VIEW query.

- **QueryTypeInsertInto**
  INSERT INTO query.

- **QueryTypeInvalid**
  Invalid query.

- **QueryTypeSelect**
  SELECT query.

- **QueryTypeSelectInto**
  SELECT INTO query.

- **QueryTypeUpdate**
  UPDATE query.

**See Also**

Scripting Reference
Scope Constants
Scopes within various mapfile components.

Values

AllLabels
All labels (used with labels components).

AllObjects
All objects (used with drawing components).

AllPixels
All pixels (used with image or surface components).

InvisiblePixels
Invisible pixels (used with image or surface components).

Selection
Selected labels, objects or pixels (used with drawing, image, labels or surface components).

See Also

Scripting Reference
**SelectMode Constants**

Selection modes.

**Values**

- **SelectAdd**
  - Addition mode.

- **SelectIntersect**
  - Intersection mode.

- **SelectInvert**
  - Inversion mode.

- **SelectReplace**
  - Replacement mode.

- **SelectSubtract**
  - Subtraction mode.

**See Also**

Scripting Reference
SubtypeAny Constants
Import subtypes.

Values

SubtypeImage
Import data as image.

SubtypeSurface
Import data as surface.

See Also

Scripting Reference
**SubtypeDb Constants**

Subtypes of DB files used in converters.

**Values**

- SubtypeDbParadox3x
  - Paradox 3.x.

- SubtypeDbParadox4x
  - Paradox 4.x.

- SubtypeDbParadox5x
  - Paradox 5.x.

**See Also**

Scripting Reference
**SubtypeDbf Constants**

Subtypes of DBF files used in converters.

**Values**

SubtypeDbfFoxPro
FoxPro.

SubtypeDbfDBase3
dBase III.

SubtypeDbfDBase4
dBase IV.

SubtypeDbfDBase5
dBase V.

**See Also**

Scripting Reference
SubtypeMdb Constants
Subtypes of MDB files used in converters.

Values

SubtypeMdbAccess2000

SubtypeMdbAccess97
Access 97.

See Also

Scripting Reference
**SubtypeTaif Constants**
Subtypes of TAIF files used in converters.

**Values**

SubtypeTaifLevel1
TAIF Level 1.

SubtypeTaifLevel2
TAIF Level 2.

SubtypeTaifStreetNet
TAIF StreetNet.

**See Also**

Scripting Reference
SubtypeWk Constants
Subtypes of WK files used in converters.

Values

SubtypeWk1
WK1.

SubtypeWk3
WK3.

SubtypeWk4
WK4.

See Also

Scripting Reference
SubtypeXls Constants
Subtypes of XLS files used in converters.

Values

SubtypeXls30
Excel 3.0.

SubtypeXls40
Excel 4.0

SubtypeXls50
Excel 5.0.

SubtypeXls80
Excel 97

See Also

Scripting Reference
**TiffCompression Constants**

TIFF compression options used in converters.

**Values**

- **TiffCompressionJpeg**
  JPEG compression.

- **TiffCompressionLzw**
  LZW compression.

- **TiffCompressionNone**
  No compression.

- **TiffCompressionPackBits**
  PackBits compression.

- **TiffCompressionZip**
  ZIP (deflate) compression.

**See Also**

Scripting Reference
TransferRuleDiv Constants
1 to N transfer rules used in table columns.

Values

TransferDivNone
None.

TransferDivCopy
Copy.

TransferDivEqual
Equal.

TransferDivProportional
Proportional (requires a column).

TransferDivSample
Sample.

See Also

Scripting Reference
TransferRuleMul Constants
N to 1 transfer rules used in table columns.

Values

TransferMulNone
None.

TransferMulAverage
Average.

TransferMulCopy
Copy.

TransferMulCount
Count.

TransferMulMaximum
Maximum.

TransferMulMedian
Median.

TransferMulMinimum
Minimum.

TransferMulSample
Sample.

TransferMulSum
Sum.

See Also

Scripting Reference
**TurnStyle Constants**  
Turn report styles (used with routes).

**Values**

- **TurnStyleNone**  
  No turns.

- **TurnStyleAbsolute**  
  "Turn East" report style.

- **TurnStyleBoth**  
  "Turn Right (East)" report style.

- **TurnStyleRelative**  
  "Turn Right" report style.

**See Also**

- **Analyzer Object**  
  Scripting Reference
**ValueType Constants**

Pixel value types used in image converters.

**Values**

**ValueTypeBit**

1 bit.

**ValueTypeFloat32**

Single precision floating-point number.

**ValueTypeFloat64**

Double precision floating-point number.

**ValueTypeInt16**

16 bit integer.

**ValueTypeInt32**

32 bit integer.

**ValueTypeInt64**

64 bit integer.

**ValueTypeInt8**

8 bit integer.

**ValueTypeUInt16**

Unsigned 16 bit integer.

**ValueTypeUInt32**

Unsigned 32 bit integer.

**ValueTypeUInt64**

Unsigned 64 bit integer.

**ValueTypeUInt8**

Unsigned 8 bit integer.

**See Also**

Scripting Reference
**WindowAlign Constants**
Alignment options for north arrow, scale bar and legend.

**Values**

WindowAlignBottomLeft
Align to bottom left corner of visible area.

WindowAlignBottomRight
Align to bottom right corner of visible area.

WindowAlignNone
No alignment.

WindowAlignTopLeft
Align to top left corner of visible area.

WindowAlignTopRight
Align to top right corner of visible area.

**See Also**
Scripting Reference

Objects

Analyzer Object
Performs spatial analysis and transforms.

Properties

Application Application
Returns application. Read only.

Parent Parent
Returns parent object. Read only.

Methods for Drawings / General Methods

ObjectSet AttachTo(Component context, ObjectSet target, ObjectSet source)
Attaches objects in target object set to objects in source object set using the coordinate system and the location precision of the context component.

ObjectSet AttachToSelf(Component context, ObjectSet target)
Attaches objects in target object set to itself using the coordinate system and the location precision of the context component.

ObjectSet BorderBuffers(Component context, Drawing drawing, ObjectSet target, Number size, String unit)
Creates border buffers of specified size for objects in target object set using the coordinate system and the location precision of the context component and places the new objects into specified drawing. Unit parameter is optional and can be a Unit object or a unit name.

ObjectSet Boundaries(Component context, Drawing drawing, ObjectSet target)
Creates boundaries for objects in target object set using the coordinate system and the location precision of the context component and places the new objects into specified drawing.

ObjectSet BoundedAreas(Component context, Drawing drawing, ObjectSet target)
Creates areas bounded by objects in target object set using the coordinate system and the location precision of the context component and places the new objects into specified drawing.

ObjectSet BoundingBoxes(Component context, Drawing drawing, ObjectSet target)
Creates bounding boxes of objects in target object set using the coordinate system and the location precision of the context component and places the new objects into specified drawing.

ObjectSet Buffers(Component context, Drawing drawing, ObjectSet target, Number size, String unit)
ObjectSet Buffers(Component context, Drawing drawing, ObjectSet target, Number size, Unit unit)
Creates buffers of specified size for objects in target object set using the coordinate system and the location precision of the context component and places the new objects into specified drawing. Unit parameter is optional and can be a Unit object or a unit name.

ObjectSet Centroids(Component context, Drawing drawing, ObjectSet target)
Creates centroids of objects in target object set using the coordinate system and the location precision of the context component and places the new objects into specified drawing.

ObjectSet CentroidsBox(Component context, Drawing drawing, ObjectSet target)

Creates box centroids (centers of bounding boxes) of objects in target object set using the coordinate system and the location precision of the context component and places the new objects into specified drawing.

ObjectSet CentroidsInner(Component context, Drawing drawing, ObjectSet target)

Creates inner centroids of areas (centroids which are guaranteed to stay within areas) in target object set using the coordinate system and the location precision of the context component and places the new objects into specified drawing.

ObjectSet CentroidsWeight(Component context, Drawing drawing, ObjectSet target)

Creates weight centroids of objects in target object set using the coordinate system and the location precision of the context component and places the new objects into specified drawing.

ObjectSet ClipIntersect(Component context, ObjectSet target, ObjectSet source)

Intersects objects in target object set with objects in source object set using the coordinate system and the location precision of the context component.

ObjectSet ClipSubtract(Component context, ObjectSet target, ObjectSet source)

Subtracts objects in source object set from objects in target object set using the coordinate system and the location precision of the context component.

ObjectSet Clusters(Component context, Drawing drawing, ObjectSet target, Number threshold)

Connects objects in target object set into clusters with specified threshold using the coordinate system and the location precision of the context component and places the new objects into specified drawing.

ObjectSet ConstrainedTriangulation(Component context, Drawing drawing, ObjectSet target)

Creates constrained triangulation of all objects in target object set using the coordinate system and the location precision of the context component and places the new objects into specified drawing.

ObjectSet ConstrainedTriangulationAreas(Component context, Drawing drawing, ObjectSet target)

Creates constrained triangulation of all objects in target object set using the coordinate system and the location precision of the context component and creates an area object for each triangle in the specified drawing.

ObjectSet ConstrainedTriangulationLines(Component context, Drawing drawing, ObjectSet target)

Creates constrained triangulation of all objects in target object set using the coordinate system and the location precision of the context component and creates a line object for each triangulation edge in the specified drawing.

AnalyzerValueSet ClustersAssoc(Component context, ObjectSet target, Number threshold)

Connects objects in target object set into clusters with specified threshold using the coordinate system and the location precision of the context component and returns a set of resulting connections.

ObjectSet ClustersZahn(Component context, Drawing drawing, ObjectSet target, Number threshold)

Connects objects in target object set into Zahn clusters with specified threshold using the coordinate system and the location precision of the context component and places the new objects into specified drawing.

AnalyzerValueSet ClustersZahnAssoc(Component context, ObjectSet target, Number threshold)

Connects objects in target object set into Zahn clusters with specified threshold using the coordinate system and the location precision of the context component and returns a set of resulting connections.

ObjectSet CommonBoundingBox(Component context, Drawing drawing, ObjectSet target)

Creates common bounding box of all objects in target object set using the coordinate system and the location precision of the context component and places the new object into specified drawing.

ObjectSet CommonBuffer(Component context, Drawing drawing, ObjectSet target, Number size, String unit)
ObjectSet CommonBuffer(Component context, Drawing drawing, ObjectSet target, Number size, Unit unit)
Creates common buffer of specified size of all objects in target object set using the coordinate system and the location precision of the context component and places the new object into specified drawing. Unit parameter is optional and can be a Unit object or a unit name.

ObjectSet CommonCentroid(Component context, Drawing drawing, ObjectSet target)
Creates common centroid of all objects in target object set using the coordinate system and the location precision of the context component and places the new object into specified drawing.

ObjectSet CommonEnclosingCircle(Component context, Drawing drawing, ObjectSet target)
Creates common enclosing circle of all objects in target object set using the coordinate system and the location precision of the context component and places the new object into specified drawing.

ObjectSet CommonEnclosingRectangle(Component context, Drawing drawing, ObjectSet target)
Creates common enclosing rectangle of all objects in target object set using the coordinate system and the location precision of the context component and places the new object into specified drawing.

ObjectSet ConvexHull(Component context, Drawing drawing, ObjectSet target)
Creates convex hull of all objects in target object set using the coordinate system and the location precision of the context component and places the new object into specified drawing.

ObjectSet Decompose(Component context, ObjectSet target)
Decompose objects in target object set using the coordinate system and the location precision of the context component.

ObjectSet DecomposeToConvexParts(Component context, ObjectSet target)
Decompose objects in target object set into convex parts using the coordinate system and the location precision of the context component.

ObjectSet DecomposeToTriangles(Component context, ObjectSet target)
Decompose objects in target object set into triangles using the coordinate system and the location precision of the context component.

ObjectSet DistanceNetwork(Component context, Drawing drawing, ObjectSet target, Number threshold, String unit)
ObjectSet DistanceNetwork(Component context, Drawing drawing, ObjectSet target, Number threshold, Unit unit)
Connects objects in target object set into distance network with specified distance threshold using the coordinate system and the location precision of the context component and places the new objects into specified drawing. Unit parameter is optional and can be a Unit object or a unit name.

AnalyzerValueSet DistanceNetworkAssoc(Component context, ObjectSet target, Number threshold, String unit)
AnalyzerValueSet DistanceNetworkAssoc(Component context, ObjectSet target, Number threshold, Unit unit)
Connects objects in target object set into distance network with specified distance threshold using the coordinate system and the location precision of the context component and returns a set of resulting connections. Unit parameter is optional and can be a Unit object or a unit name.

AnalyzerValueSetColumn District (ObjectSet areas, Column balance, Number districts, Number threshold)
Splits set of areas to a given number of districts balancing given column. Column can be supplied directly or by its ID or name. Threshold is a maximum possible distance in native coordinate system units at which the areas can connect and is optional. Returns a set of associations between areas and their district numbers.

AnalyzerValueSetColumn DistrictAdv (ObjectSet areas, Column balance, String districts, Column startWith)
Splits set of areas to districts balancing given column. Column and StartWith columns can be supplied directly or by their IDs or names. Districts contains relative weights of districts separated by commas. There can be no negative or zero weights and there should be at least two weights. StartWith contains initial district information and is optional. Returns a set of associations between areas and their district numbers. Requires Business Tools extension.
AnalyzerValueSetColumn DistrictAdvGeom(ObjectSet areas, Column balance, String districts, Column startWith, Number balancing, Number compactness)
Splits set of areas to districts balancing given column. Balance and StartWith columns can be supplied directly or by their IDs or names. Districts contains relative weights of districts separated by commas. There can be no negative or zero weights and there should be at least two weights. StartWith contains initial district information and is optional. Balancing and Compactness contain relative preferences for more balanced districts or more compact districts. Returns a set of associations between areas and their district numbers. Requires BusinessTools extension.

Geom DriveTimeZone(ObjectSet roads, Object center, Column length, Column speed, Number speedOverLand, Number time)
Geom DriveTimeZone(ObjectSet roads, ObjectSet centers, Column length, Column speed, Number speedOverLand, Number time)
Creates a drive-time zone using a road network and a single center or a set of centers. Length and Speed columns can be supplied directly or by their IDs or names. SpeedOverLand is a speed of movement “across” the roads. Requires Business Tools extension.

Geom DriveTimeZoneBuffer(ObjectSet roads, Object center, Column length, Column speed, Number time, Number buffer, Unit bufferUnit)
Geom DriveTimeZoneBuffer(ObjectSet roads, ObjectSet centers, Column length, Column speed, Number time, Number buffer, Unit bufferUnit)
Creates a drive-time buffer using a road network and a single center or a set of centers. Length and Speed columns can be supplied directly or by their IDs or names. Requires Business Tools extension.

Geom DriveTimeZoneConvex(ObjectSet roads, Object center, Column length, Column speed, Number time)
Geom DriveTimeZoneConvex(ObjectSet roads, ObjectSet centers, Column length, Column speed, Number time)
Creates a drive-time hull using a road network and a single center or a set of centers. Length and Speed columns can be supplied directly or by their IDs or names. Requires Business Tools extension.

ObjectSet EnclosingCircles(Component context, Drawing drawing, ObjectSet target)
Creates enclosing circles for objects in target object set using the coordinate system and the location precision of the context component and places the new objects into specified drawing.

ObjectSet EnclosingRectangles(Component context, Drawing drawing, ObjectSet target)
Creates enclosing rectangles for objects in target object set using the coordinate system and the location precision of the context component and places the new objects into specified drawing.

ObjectSet Explode(Component context, ObjectSet target)
Decompose lines in target object set to segments using the coordinate system and the location precision of the context component.

ObjectSet FarthestNeighbor(Component context, Drawing drawing, ObjectSet target, ObjectSet source)
Connects each object in source object set with its farthest neighbor in target object set using the coordinate system and the location precision of the context component and places the new objects into specified drawing.

AnalyzerValueSet FarthestNeighborAssoc(Component context, ObjectSet target, ObjectSet source)
Connects each object in source object set with its farthest neighbor in target object set using the coordinate system and the location precision of the context component and returns a set of resulting connections.

ObjectSet FarthestNeighborSymm(Component context, Drawing drawing, ObjectSet target, ObjectSet source)
Connects each object in source object set with its symmetric farthest neighbor in target object set using the coordinate system and the location precision of the context component and places the new objects into specified drawing.

AnalyzerValueSet FarthestNeighborSymmAssoc(Component context, ObjectSet target, ObjectSet source)
Connects each object in source object set with its symmetric farthest neighbor in target object set using the coordinate system and the location precision of the context component and returns a set of resulting connections.
ObjectSet FarthestPair(Component context, Drawing drawing, ObjectSet target)
Connects farthest pair of objects in target object set using the coordinate system and the location precision of
the context component and places the new object into specified drawing.

AnalyzerValueSet FarthestPairAssoc(Component context, ObjectSet target)
Connects farthest pair of objects in target object set using the coordinate system and the location precision of
the context component and returns the resulting connection.

ObjectSet FlipHorizontally(Component context, ObjectSet target)
Flips objects in target object set horizontally using the coordinate system and the location precision of the
context component.

ObjectSet FlipVertically(Component context, ObjectSet target)
Flips objects in target object set vertically using the coordinate system and the location precision of the context
component.

ObjectSet GabrielNetwork(Component context, Drawing drawing, ObjectSet target)
Connects objects in target object set into Gabriel network using the coordinate system and the location precision
of the context component and places the new objects into specified drawing.

AnalyzerValueSet GabrielNetworkAssoc(Component context, ObjectSet target)
Connects objects in target object set into Gabriel network using the coordinate system and the location precision
of the context component and returns a set of resulting connections.

ObjectSet InnerBuffers(Component context, Drawing drawing, ObjectSet target, Number size, String unit)
ObjectSet InnerBuffers(Component context, Drawing drawing, ObjectSet target, Number size, Unit unit)
Creates inner buffers of specified size for objects in target object set using the coordinate system and the
location precision of the context component and places the new objects into specified drawing. Unit parameter
is optional and can be a Unit object or a unit name.

ObjectSet IntersectLines(Component context, ObjectSet target)
Intersects lines in target object set using the coordinate system and the location precision of the context
component.

ObjectSet IntersectionPoints(Component context, Drawing drawing, ObjectSet target)
Creates intersection points of lines in target object set using the coordinate system and the location precision of
the context component and places the new objects into specified drawing.

ObjectSet JoinLines(Component context, Drawing drawing, ObjectSet target)
Joins lines in target object set using the coordinate system and the location precision of the context component
and places the new objects into specified drawing.

ObjectSet MoveHorizontally(Component context, ObjectSet target, Number size, String unit)
ObjectSet MoveHorizontally(Component context, ObjectSet target, Number size, Unit unit)
Moves objects in target object set horizontally by specified distance using the coordinate system and the location
precision of the context component. Unit parameter is optional and can be a Unit object or a unit name.

ObjectSet MoveVertically(Component context, ObjectSet target, Number size, String unit)
ObjectSet MoveVertically(Component context, ObjectSet target, Number size, Unit unit)
Moves objects in target object set vertically by specified distance using the coordinate system and the location
precision of the context component. Unit parameter is optional and can be a Unit object or a unit name.

ObjectSet NearestNeighbor(Component context, Drawing drawing, ObjectSet target, ObjectSet source)
Connects each object in source object set with its nearest neighbor in target object set using the coordinate system and the location precision of the context component and places the new objects into specified drawing.

AnalyzerValueSet NearestNeighborAssoc(Component context, ObjectSet target, ObjectSet source)

Connects each object in source object set with its nearest neighbor in target object set using the coordinate system and the location precision of the context component and returns a set of resulting connections.

ObjectSet NearestNeighborSymm(Component context, Drawing drawing, ObjectSet target, ObjectSet source)

Connects each object in source object set with its symmetric nearest neighbor in target object set using the coordinate system and the location precision of the context component and places the new objects into specified drawing.

AnalyzerValueSet NearestNeighborSymmAssoc(Component context, ObjectSet target, ObjectSet source)

Connects each object in source object set with its symmetric nearest neighbor in target object set using the coordinate system and the location precision of the context component and returns a set of resulting connections.

ObjectSet NearestPair(Component context, Drawing drawing, ObjectSet target)

Connects nearest pair of objects in target object set using the coordinate system and the location precision of the context component and places the new object into specified drawing.

AnalyzerValueSet NearestPairAssoc(Component context, ObjectSet target)

Connects nearest pair of objects in target object set using the coordinate system and the location precision of the context component and returns the resulting connection.

ObjectSet NodePoints(Component context, Drawing drawing, ObjectSet target)

Creates node points for lines in target object set using the coordinate system and the location precision of the context component and places the new objects into specified drawing.

ObjectSet NormalizeMetric(Component context, ObjectSet target)

Normalizes metric of objects in target object set using the coordinate system and the location precision of the context component.

ObjectSet NormalizeTopology(Component context, ObjectSet target)

Normalizes topology of objects in target object set using the coordinate system and the location precision of the context component.

Route OptimalRoute(ObjectSet locations, ObjectSet roads, Column length, Column speed, Boolean split, Object start, Object finish)

Locates optimal route between given locations using given roads. There should be at least two locations and at least one road. Length and Speed columns can be supplied directly or by their IDs or names. Split controls whether the roads are split at intersections and locations (the default) or not. Start and finish parameters are optional. Returns a route. Requires Business Tools extension.

ObjectSet Points(Component context, Drawing drawing, ObjectSet target)

Creates inflection points of objects in target object set using the coordinate system and the location precision of the context component and places the new objects into specified drawing.

ObjectSet RelativeNeighborhoodNetwork(Component context, Drawing drawing, ObjectSet target)

Connects objects in target object set into relative neighborhood network using the coordinate system and the location precision of the context component and places the new objects into specified drawing.

AnalyzerValueSet RelativeNeighborhoodNetworkAssoc(Component context, ObjectSet target)

Connects objects in target object set into relative neighborhood network using the coordinate system and the location precision of the context component and returns a set of resulting connections.

ObjectSet RemoveDuplicates(Component context, ObjectSet target)
Removes duplicate objects in target object set using the coordinate system and the location precision of the
correct component.

ObjectSet ReverseLines(Component context, ObjectSet target)
Reverses direction of lines in target object set using the coordinate system and the location precision of the
correct component.

ObjectSet Rotate(Component context, ObjectSet target, Number angle)
Rotates objects in target object set by specified amount of degrees using the coordinate system and the location
precision of the correct component.

ObjectSet Scale(Component context, ObjectSet target, Number scale)
Scales objects in target object set by specified factor using the coordinate system and the location precision of the
correct component.

ObjectSet ScaleHorizontally(Component context, ObjectSet target, Number scale)
Scales objects in target object set horizontally by specified factor using the coordinate system and the location
precision of the correct component.

ObjectSet ScaleVertically(Component context, ObjectSet target, Number scale)
Scales objects in target object set vertically by specified factor using the coordinate system and the location
precision of the correct component.

ObjectSet Segments(Component context, ObjectSet target, Number segments)
Splits straight line segments in objects in target object set into specified number of subsegments using the
coordinate system and the location precision of the correct component.

ObjectSet SelectAdjacentTo(Component context, ObjectSet target, ObjectSet source)
Selects objects in target object set adjacent to objects in source object set using the coordinate system and the
location precision of the correct component.

ObjectSet SelectContainedWithin(Component context, ObjectSet target, ObjectSet source)
Selects objects in target object set contained within objects in source object set using the coordinate system and the
location precision of the correct component.

ObjectSet SelectContaining(Component context, ObjectSet target, ObjectSet source)
Selects objects in target object set containing objects in source object set using the coordinate system and the
location precision of the correct component.

ObjectSet SelectCriticalServiceCenter(Component context, ObjectSet target, Number candidates)
Selects specified number of candidates for a critical service center for objects in target object set using the
coordinate system and the location precision of the correct component.

ObjectSet SelectEuclideanPointCoverage(Component context, ObjectSet target, Number threshold, String unit)

ObjectSet SelectEuclideanPointCoverage(Component context, ObjectSet target, Number threshold, Unit unit)
Selects minimum Euclidean point coverage with given distance threshold for objects in target object set using the
coordinate system and the location precision of the correct component. Unit parameter is optional and can be a Unit object or a unit name.

ObjectSet SelectIntersecting(Component context, ObjectSet target, ObjectSet source)
Selects objects in target object set intersecting objects in source object set using the coordinate system and the
location precision of the correct component.

ObjectSet SelectLineCoverage(Component context, ObjectSet target)
Selects minimum line coverage for objects in target object set using the coordinate system and the location
precision of the correct component.
ObjectSet SelectNCriticalServiceCenters(Component context, ObjectSet target, Number centers)
Selects locations for specified number of critical service centers for objects in target object set using the coordinate system and the location precision of the context component.

ObjectSet SelectNServiceCenters(Component context, ObjectSet target, Number centers)
Selects locations for specified number of regular service centers for objects in target object set using the coordinate system and the location precision of the context component.

ObjectSet SelectPointCoverage(Component context, ObjectSet target)
Selects minimum point coverage for objects in target object set using the coordinate system and the location precision of the context component.

ObjectSet SelectServiceCenter(Component context, ObjectSet target, Number candidates)
Selects specified number of candidates for a regular service center for objects in target object set using the coordinate system and the location precision of the context component.

ObjectSet SelectShortestPath(Component context, ObjectSet target, Object beg, Object end)
Selects shortest path between specified objects via objects in target object set using the coordinate system and the location precision of the context component.

ObjectSet SelectSpanningTree(Component context, ObjectSet target)
Selects minimum spanning tree for objects in target object set using the coordinate system and the location precision of the context component.

ObjectSet SelectTouching(Component context, ObjectSet target, ObjectSet source)
Selects objects in target object set touching objects in source object set using the coordinate system and the location precision of the context component.

ObjectSet ShapeHull(Component context, Drawing drawing, ObjectSet target, Number resolution)
Creates shape hull of specified resolution for all objects in target object set using the coordinate system and the location precision of the context component and places the new object into specified drawing.

ObjectSet SpanningTree(Component context, Drawing drawing, ObjectSet target)
Connects objects in target object set into minimum Euclidean spanning tree using the coordinate system and the location precision of the context component and places the new objects into specified drawing.

AnalyzerValueSet SpanningTreeAssoc(Component context, ObjectSet target)
Connects objects in target object set into minimum Euclidean spanning tree using the coordinate system and the location precision of the context component and returns a set of resulting connections.

ObjectSet Spline(Component context, ObjectSet target, Number midpoints)
Smoothes lines and areas in target object set by converting each segment into a spline with the given number of midpoints. Uses the coordinate system and the location precision of the context component.

ObjectSet Split(Component context, Drawing drawing, ObjectSet target, ObjectSet source)
Splits objects in target object set with objects in source object set using the coordinate system and the location precision of the context component and places the new objects into specified drawing.

ObjectSet Triangulation(Component context, Drawing drawing, ObjectSet target)
Creates triangulation of all objects in target object set using the coordinate system and the location precision of the context component and places the new objects into specified drawing.

ObjectSet TriangulationAreas(Component context, Drawing drawing, ObjectSet target)
Creates triangulation areas of all objects in target object set using the coordinate system and the location precision of the context component and places the new objects into specified drawing.
ObjectSet TriangulationLines(Component context, Drawing drawing, ObjectSet target)
Creates triangulation lines of all objects in target object set using the coordinate system and the location precision of the context component and places the new objects into specified drawing.

ObjectSet Union(Component context, Drawing drawing, ObjectSet target)
Unions all areas in target object set using the coordinate system and the location precision of the context component and places the new object into specified drawing.

ObjectSet Voronoi(Component context, Drawing drawing, ObjectSet target)
Creates Voronoi diagram of all objects in target object set using the coordinate system and the location precision of the context component and places the new objects into specified drawing.

ObjectSet VoronoiAreas(Component context, Drawing drawing, ObjectSet target)
Creates Voronoi areas of all objects in target object set using the coordinate system and the location precision of the context component and places the new objects into specified drawing.

ObjectSet VoronoiLines(Component context, Drawing drawing, ObjectSet target)
Creates Voronoi lines of all objects in target object set using the coordinate system and the location precision of the context component and places the new objects into specified drawing.

ObjectSet VoronoiPoints(Component context, Drawing drawing, ObjectSet target)
Creates Voronoi points of all objects in target object set using the coordinate system and the location precision of the context component and places the new objects into specified drawing.

Methods for Images and Surfaces

AddMargin(Image component, Number pixels)
AddMargin(Surface component, Number pixels)
Add margins of specified size to image or surface component.

AddNoise(PixelSet pixelSet, Number amount)
Add specified amount of noise to a set of image pixels.

AddNoiseMono(PixelSet pixelSet, Number amount)
Add specified amount of monochromatic noise to a set of image pixels.

AutoContrast(PixelSet pixelSet)
Autocontrasts a set of image pixels.

AutoLevels(PixelSet pixelSet)
Autolevels a set of image pixels.

Blur(PixelSet pixelSet, Number amount)
Filters a set of image or surface pixels with Blur. Amount parameter is optional.

Brightness(PixelSet pixelSet, Number amount)
Adjusts brightness of a set of image pixels by specified amount.

Contrast(PixelSet pixelSet, Number amount)
Adjusts contrast of a set of image pixels by specified amount.

CropMargin(Image component, Number pixels)
CropMargin(Surface component, Number pixels)
Crops margins of specified size from image or surface component.
Crop(PixelSet pixelSet)
Crops image or surface to given set of pixels.

Desaturate(PixelSet pixelSet)
Desaturates a set of image pixels.

DifferenceE(PixelSet pixelSet)
Filters a set of image or surface pixels with Difference East.

DifferenceN(PixelSet pixelSet)
Filters a set of image or surface pixels with Difference North.

DifferenceNE(PixelSet pixelSet)
Filters a set of image or surface pixels with Difference North East.

DifferenceNW(PixelSet pixelSet)
Filters a set of image or surface pixels with Difference North West.

DifferenceS(PixelSet pixelSet)
Filters a set of image or surface pixels with Difference South.

DifferenceSE(PixelSet pixelSet)
Filters a set of image or surface pixels with Difference South East.

DifferenceSW(PixelSet pixelSet)
Filters a set of image or surface pixels with Difference South West.

DifferenceW(PixelSet pixelSet)
Filters a set of image or surface pixels with Difference West.

Diffuse(PixelSet pixelSet, Number amount)
Diffuses colors in a set of image pixels by specified amount.

Equalize(PixelSet pixelSet, Number amount)
Equalizes colors in a set of image pixels by specified amount.

FlipPixelsHorizontally(PixelSet pixelSet)
Flips a set of image or surface pixels horizontally.

FlipPixelsVertically(PixelSet pixelSet)
Flips a set of image or surface pixels vertically.

Gamma(PixelSet pixelSet, Number amount)
Adjusts gamma of a set of image pixels by specified amount.

GaussianBlur(PixelSet pixelSet, Number radius)
Applies gaussian blur of specified radius to a set of image pixels.

Grayscale(PixelSet pixelSet)
Grayscale a set of image pixels.

HighPass1(PixelSet pixelSet)
Filters a set of image or surface pixels with High Pass 1.

HighPass2(PixelSet pixelSet)
Filters a set of image or surface pixels with High Pass 2.

HighPass3(PixelSet pixelSet)
Filters a set of image or surface pixels with High Pass 3.

Interpolate(PixelSet pixelSet, Number radius)
Interpolates invisible pixels in a set of surface pixels with given radius. Radius parameter is optional.

InterpolateRow(PixelSet pixelSet)
Interpolates invisible pixels in a set of surface pixels going by rows.

Invert(PixelSet pixelSet, Number level)
Inverts a set of image or surface pixels at specified level. Level parameter is optional.

Laplace1(PixelSet pixelSet)
Filters a set of image or surface pixels with Laplace 1.

Laplace2(PixelSet pixelSet)
Filters a set of image or surface pixels with Laplace 2.

LowPass1(PixelSet pixelSet)
Filters a set of image or surface pixels with Low Pass 1.

LowPass2(PixelSet pixelSet)
Filters a set of image or surface pixels with Low Pass 2.

LowPass3(PixelSet pixelSet)
Filters a set of image or surface pixels with Low Pass 3.

MedianCross(PixelSet pixelSet)
Filters a set of image or surface pixels with Median Cross.

MedianSquare(PixelSet pixelSet)
Filters a set of image or surface pixels with Median Square.

MedianSquare5(PixelSet pixelSet)
Filters a set of image or surface pixels with Median Square 5.

MotionBlurDiagonal1(PixelSet pixelSet, Number distance)
Applies (forward) diagonal motion blur of specified distance to a set of image pixels.

MotionBlurDiagonal2(PixelSet pixelSet, Number distance)
Applies (backward) diagonal motion blur of specified distance to a set of image pixels.

MotionBlurHorizontal(PixelSet pixelSet, Number distance)
Applies horizontal motion blur of specified distance to a set of image pixels.

MotionBlurVertical(PixelSet pixelSet, Number distance)
Applies vertical motion blur of specified distance to a set of image pixels.
Posterize(PixelSet pixelSet, Number levels)
Posterizes a set of image pixels to specified number of levels.

RotatePixels(Image component, Number angle)
RotatePixels(Surface component, Number angle)
Rotates image or surface to specified angle (in degrees).

Sharpen(PixelSet pixelSet, Number amount)
Filters a set of image or surface pixels with Sharpen. Amount parameter is optional.

SharpenMore(PixelSet pixelSet)
Filters a set of image or surface pixels with Sharpen More.

Threshold(PixelSet pixelSet, Number level)
Thresholds a set of image pixels at given level to black and white.

ThresholdBlack(PixelSet pixelSet, Number level)
Thresholds a set of image pixels at given level to black.

ThresholdLower(PixelSet pixelSet, Number level)
Thresholds pixels lower than given level in a set of surface pixels.

ThresholdUpper(PixelSet pixelSet, Number level)
Thresholds pixels higher than given level in a set of surface pixels.

ThresholdWhite(PixelSet pixelSet, Number level)
Thresholds a set of image pixels at given level to white.

Tile(PixelSet pixelSet, Number size)
Tiles a set of image or surface pixels choosing mean pixel values.

TileMedian(PixelSet pixelSet, Number size)
Tiles a set of image or surface pixels choosing median pixel values.

Obtained From
Document (NewAnalyzer)

See Also
Scripting Reference
AnalyzerValue
AnalyzerValue Object
An association between a pair of objects.

Properties

Application Application
  Returns application. Read only.

Any Key
  Returns key object. Read only.

Any Parent
  Returns parent object. Read only.

Any Value
  Returns value object. Read only.

Obtained From

AnalyzerValueSet (Item)

See Also

Scripting Reference
AnalyzerValueSet Object
A set of associations between pairs of objects.

Properties

Application Application
   Returns application. Read only.

Number Count
   Returns number of associations in set. Read only.

AnalyzerValue Item(Number index)
   Returns association with given index. Read only.

Any Parent
   Returns parent object. Read only.

Methods

(none)

Obtained From

Analyzer (ClustersAssoc, ClustersZahnAssoc, DistanceNetworkAssoc, FarthestNeighorAssoc, FarthestNeighborSymmAssoc, FarthestPairAssoc, GabrielNetworkAssoc, NearestNeighborAssoc, NearestNeighborSymmAssoc, NearestPairAssoc, RelativeNeighborhoodNetworkAssoc, SpanningTreeAssoc)

See Also

Scripting Reference
**AnalyzerValueSetColumn Object**
A set of associations between pairs of objects.

**Properties**

Application Application
Returns application. Read only.

Number Count
Returns number of associations in set. Read only.

AnalyzerValueItem(Number index)
Returns association with given index. Read only.

Any Parent
Returns parent object. Read only.

**Methods**

SaveTo(Column column)
Saves values associated with objects to a given column. Column can be supplied directly or by its ID or name.

**Obtained From**

Analyzer (District, DistrictAdv, DistrictAdvGeom)

**See Also**

Scripting Reference
**Application Object**
Manages documents and application-wide properties.

**Properties**

Document ActiveDocument
- Returns active document. Read only.

Window ActiveWindow
- Returns active window. Read only.

Application Application
- Returns itself. Read only.

String Caption
- Sets or returns caption string. Read/write.

CoordinateSystemSet CoordinateSystemSet
- Returns available coordinate systems. Read only.

DatumSet DatumSet
- Returns available datums. Read only.

CoordinateSystem DefaultCoordinateSystem
- Returns default coordinate system for projected components. Read only.

CoordinateSystem DefaultCoordinateSystemLatLon
- Returns default coordinate system for latitude/longitude components. Read only.

Datum DefaultDatum
- Returns default datum. Read only.

Ellipsoid DefaultEllipsoid
- Returns default ellipsoid. Read only.

String DefaultFilePath
- Sets or returns default folder for mapfiles. Read/write.

Unit DefaultUnit
- Returns default unit for projected components. Read only.

Unit DefaultUnitLatLon
- Returns default unit for latitude/longitude components. Read only.

DocumentSet DocumentSet
- Returns set of open documents. Read only.

EllipsoidSet EllipsoidSet
- Returns available ellipsoids. Read only.

String FullName
- Returns fully qualified path to application executable. Read only.
Number Height
Sets or returns height of the main window in pixels. Read/write.

History History
Returns history object. Read only.

Boolean Interactive
Returns True if application is operating in interactive mode. Read only.

Number Left
Sets or returns left coordinate of the main window in pixels. Read/write.

String Name
Returns name of the application. Read only.

Any Parent
Returns itself. Read only.

String Path
Returns path to application executable. Read only.

RecentDataSourceSet RecentDataSourceSet
Returns a set of recently used data sources. Read only.

String StatusText
Sets or returns the contents of the status bar. Read/write.

String SerialNumberHash
Returns a hashed value for the serial number, for example, "BVh\...". Read only.

String SerialNumberHashHex
Returns a hashed value for the serial number in hexadecimal notation, for example, "121F\...". Read only.

Number Top
Sets or returns top coordinate of the main window in pixels. Read/write.

UnitSet UnitSet
Returns available units. Read only.

UserInterface UserInterface
Returns a user interface object that allows invoking commands and controlling elements of the user interface such as dialogs and controls. Read only.

Version Version
Returns version information. Read only.

Boolean Visible
Sets or returns visibility state of the main window (only works for scripts launched from the Manifold GUI). Read/write.

Number Width
Sets or returns width of the main window in pixels. Read/write.
WindowSet WindowSet
 Returns set of opened windows. Read only.

Methods

Help()
 Invokes help system.

String BytesToString(Object bytes)
 Converts an array of bytes into a string, automatically recognizing text encoding.

String InputBox(String prompt, String caption, String initialValue)
 Displays the input box with specified prompt and caption and waits for user input. Initial value is optional.

MessageBoxResult MessageBox(String message, String caption)
 Displays message box with specified text and optional caption.

MessageBoxResult MessageBoxEx(String message, String caption, MessageBoxType type)
 Displays message box with specified text, caption, icons and buttons. Returns the identifier of the pressed button.

BranchSet NewBranchSet()
 Creates new branch set.

Color NewColor(String name, Number red, Number green, Number blue, Number alpha)
 Creates new color with given name and RGBA components. All parameters are optional.

ColumnLookupSet NewColumnLookupSet()
 Creates new column lookup set.

ColumnSet NewColumnSet()
 Creates new set of columns.

ControlPoint NewControlPoint(String name, Number x, Number y)
 Creates new control point with given name and location.

ConverterProperty NewConverterProperty(String name, String value)
 Creates new converter property for given name and value.

CoordinateConverter NewCoordinateConverter()
 Creates new coordinate converter.

CoordinateSystem NewCoordinateSystem(String preset)
 Creates new coordinate system using name of coordinate system preset.

CoordinateSystem NewCoordinateSystemFrom(String xml)
 Creates new coordinate system from given XML string.

CoordinateSystem NewCoordinateSystemFromBinary(Object binary)
 Creates new coordinate system from binary data.

CoordinateSystem NewCoordinateSystemFromEpsg(String epsg)
Manifold® System Release 8.00 User Manual

Creates new coordinate system from given EPSG code.

CoordinateSystem NewCoordinateSystemFromFile(String path)
Creates new coordinate system from given XML file.

CoordinateSystem NewCoordinateSystemFromFileAUX(String path)
Creates new coordinate system from given ERDAS AUX file.

CoordinateSystem NewCoordinateSystemFromTextGSR(String text, Number width, Number height)
Creates new coordinate system from given GSR string. The last two parameters are the width and height of an image or surface which has the coordinate system, and are optional.

CoordinateSystem NewCoordinateSystemFromTextPRJ(String text)
Creates new coordinate system from given PRJ string.

CoordinateSystem NewCoordinateSystemFromTextWKT(String text)
Creates new coordinate system from given WKT string (same as PRJ string).

DataSourceColumnSet NewDataSourceColumnSet()
Creates new column set for specifying export or import columns.

Datum NewDatum(String name)
Creates new datum using datum name.

Document NewDocument(String name, Boolean readOnly)
Creates new document or opens document from given file.

Ellipsoid NewEllipsoid(String name)
Creates new ellipsoid using ellipsoid name.

Export NewExport(String name)
Creates new exporter using exporter name.

Geocoder NewGeocoder()
Creates new geocoder object.

Geom NewGeom(GeomType type)
Creates new geometric entity of given type.

Geom NewGeom(GeomType type, BranchSet branchSet)
Creates new geometric entity of given type and branches.

Geom NewGeom(GeomType type, Branch branch)
Creates new geometric entity of given type and single branch.

Geom NewGeom(GeomType type, PointSet pointSet)
Creates new geometric entity of given type and single branch.

Geom NewGeomFromBinary(Object binary)
Creates new geometric entity from binary data in Geometry format.

Geom NewGeomFromBinarySDE(Object binary, Number shiftX, Number shiftY, Number scale)
Creates new geometric entity from binary data in Geometry (SDE) format. Shift and scale parameters specify conversion factors from integer coordinates and are optional. If omitted, shiftX is set to 0, shiftY is set to 0, and scale is set to 1.

`Geom NewGeomFromBinarySHP(Object binary)`
Creates new geometric entity from binary data in Geometry (SHP) format.

`Geom NewGeomFromBinaryWKB(Object binary)`
Creates new geometric entity from binary data in Geometry (WKB) format.

`Geom NewGeomFromTextWKT(String text)`
Creates new geometric entity from textual data in OGC WKT format.

`Import NewImport(String name)`
Creates new importer using importer name.

`Point NewPoint(Number x, Number y)`
Creates new point with given coordinates. Both parameters are optional.

`Point NewPointFromUSNG(String usng)`
Creates a new point from USNG coordinate string.

`PointSet NewPointSet()`
Creates new set of points.

`Progress NewProgress()`
Creates new progress indicator.

`Rect NewRect(Number minX, Number minY, Number maxX, Number maxY)`
Creates new rectangle object with specified coordinates. All parameters are optional.

`Topology NewTopology()`
Creates new topology object.

`Unit NewUnit(String name)`
Creates new unit using unit name.

`View NewView(String name, Number scale, Number x, Number y)`
Creates new view with given name, scale and location.

`Quit()`
Quits application; prompts to save changes.

`Repeat()`
Repeats last undone action.

`Undo()`
Undoes last action.

**Obtained From**

No need to create (accessed through global Application object).
See Also

Scripting Reference
Branch Object
A sequence of points representing part of geometric object.

Properties

Application Application
  Returns application. Read only.

Number Area
  Returns area measured over an ellipsoid. Returned value is in square meters. Read only.

Number AreaNative
  Returns area in native units. Read only.

Rect Box
  Sets or returns bounding box. Modifying this property scales and shifts the branch. Read/write.

Point Center
  Sets or returns center of minimum enclosing circle. Modifying this property moves the branch. Read/write.

Number Length
  Returns length or perimeter measured over an ellipsoid. Returned value is in meters. Read only.

Number LengthNative
  Returns length or perimeter in native units. Read only.

Any Parent
  Returns parent object. Read only.

PointSet PointSet
  Sets or returns set of points (can accept Branch or PointSet when setting). Read/write.

Number SequentialNumber
  Returns sequential number of the branch in its parent. Read only.

Number Start
  Returns sequential number of the first point of the branch in its parent. Read only.

Methods

Copy(Branch branch)
  Copies given branch.

Boolean IsEmpty()
  Returns True if branch contains no points and False otherwise.

Boolean IsFirst()
  Returns True if branch is the first branch in its parent and False otherwise.

Boolean IsLast()
  Returns True if branch is the last branch in its parent and False otherwise.
Obtained From

BranchSet (Item, LastAdded)

See Also

Scripting Reference
BranchSet Object
Set of branches representing geometric object.

Properties

Application Application
Returns application. Read only.

Rect Box
Sets or returns bounding box. Modifying this property scales and shifts the branch set. Read/write.

Point Center
Sets or returns center of minimum enclosing circle. Modifying this property moves the branch set. Read/write.

Number Count
Returns number of branches in set. Read only.

Branch Item(Number index)
Sets or returns branch with given index (can accept Branch or PointSet when setting). Read/write.

Branch LastAdded
Returns last added branch if any. Read only.

Any Parent
Returns parent object. Read only.

Methods

Add(Branch branch)
Appends branch to branch set.

Add(PointSet pointSet)
Creates new branch from given point set and appends it to branch set.

Copy(BranchSet branchSet)
Copies given branch set.

Remove(Number index)
Removes branch with given index from branch set.

RemoveAll()
Removes all branches.

Obtained From

Application (NewBranchSet)
Geom (BranchSet)
Label (BranchSet)
Object (BranchSet)

See Also
Chart Object
Chart component.

Properties

Application Application
Returns application. Read only.

String Description
Sets or returns component description. Read/write.

Document Document
Returns owner document. Read only.

Folder Folder
Sets or returns containing folder. Read/write.

ColumnSet Groups
Returns set of columns used for grouping. Read only.

Number ID
Returns unique component ID. Read only.

Boolean LogarithmicScale
Sets or returns logarithmic scale option. Read/write.

String Name
Sets or returns component name. Read/write.

String Note
Sets or returns notes associated with the component. Read/write.

ComponentSet OwnedComponentSet
Returns set of components bound to this component. Read only.

ComponentSet OwnedLayoutSet
Returns set of layouts bound to this component. Read only.

Component Owner
Returns owner component if any. Read only.

Any Parent
Returns parent object. Read only.

Border PrintBorder
Sets or returns border used for printing. Read/write.

Boolean PrintLegend
Sets or returns legend printing option. Read/write.

ColumnSet Series
Returns set of displayed columns. Read only.
ComponentType Type
Returns component type. Read only.

String TypeName
Returns textual representation of the component type. Read only.

Methods

Copy()
Copies component into the Clipboard.

Open()
Opens component.

OpenInNewWindow()
Opens component in new window.

Print(Boolean useDialog)
Prints component.

RenderTo(String name, Number width, Number height)
Render component into an image with given name, width and height.

Obtained From

Document (NewChart)

See Also

Scripting Reference
Color Object  
Color value.

Properties

Number Alpha
Returns value of the alpha channel from 0 to 255. 0 means that the color is completely opaque and 255 means that the color is completely transparent. Read only.

Application Application
Returns application. Read only.

Number Blue
Returns intensity of the blue channel from 0 to 255. Read only.

Number Green
Returns intensity of the green channel from 0 to 255. Read only.

Number Hue
Returns value of the hue from 0 to 255. Read only.

Number Intensity
Returns absolute value of the intensity from 0 to 255. Read only.

Number Lightness
Returns value of the lightness from 0 to 255. Read only.

String Name
Returns color name. Read only.

Any Parent
Returns parent object. Read only.

Number Red
Returns intensity of the red channel from 0 to 255. Read only.

Number Saturation
Returns value of the saturation from 0 to 255. Read only.

Methods

Color Darker()
Creates a darker version of the color.

Color Lighter()
Creates a lighter version of the color.

Obtained From

Application (NewColor)
ColorSet (Item)
Drawing (BackgroundColor)
Elevation (BackgroundColor)
Image (BackgroundColor)
Label (BackColor, ForeColor)
Labels (BackgroundColor)
**Layout** (BackgroundColor)
LayoutEntry (BackgroundColor, BackgroungCustomColor, BorderColor, Color)
Map (BackgroundColor)
Object (BackColor, ForeColor)
Pixel (Color)
Profile (BackgroundColor)
RichForm (BackColor, ForeColor)
Surface (BackgroundColor)
Terrain (BackgroundColor, FogColor, NorthArrowBackgroundColor, NorthArrowColor, WaterColor)
Theme (BackgroundColor)

**See Also**

Scripting Reference
ColorSet Object
Set of color values.

Properties

Application Application
  Returns application. Read only.

Number Count
  Returns number of colors in set. Read only.

Color Item(Number index)
  Returns color with given index. Read only.

Any Parent
  Returns parent object. Read only.

Obtained From

Palette (ColorSet, Selection)
PaletteSelectionSet (Item)

See Also

Scripting Reference
**Column Object**  
Table column.

**Properties**

**Application Application**  
Returns application. Read only.

**Boolean AutoRefresh**  
Sets or returns autorefresh flag of the active column. Read/write.

**ColumnCategory Category**  
Returns column category. Read only.

**Number CodePage**  
Sets or returns the code page of an ANSI text column. Read/write.

**ComputationMode ComputationMode**  
Sets or returns column computation mode (active columns only). Read/write.

**Variant DefaultValue**  
Sets or returns default value. Read/write.

**String DefaultValueText**  
Returns default value in textual form. Read only.

**DSSQuery DSSQuery**  
Returns DSS query of the rank column. Read only.

**ColumnFormat Format**  
Returns column formatting options. Read only.

**String Function**  
Sets or returns function name of the active column. Read/write.

**Number ID**  
Returns unique column ID. Read only.

**Boolean Identity**  
Sets or returns identity flag. Read/write.

**ColumnLookupSet LookupValues**  
Returns values of the lookup column. Read only.

**String Name**  
Sets or returns column name. Read/write.

**Boolean Nullable**  
Sets or returns nullable flag of the linked column. Read/write.

**Column OriginColumn**  
Sets or returns origin column of the mapped column. Read/write.
Component Owner
Returns table component containing this column if any. Read only.

Any Parent
Returns parent object. Read only.

Boolean PrimaryKey
Sets or returns primary key flag of the linked column. Read/write.

Number Size
Sets or returns column size. Read/write.

Table Table
Returns containing table if any. Read only.

TransferRuleDiv TransferDiv
Sets or returns 1 to N transfer rule.

Column TransferDivColumn
Sets or returns a column used with "proportional" 1 to N transfer rule.

TransferRuleMul TransferMul
Sets or returns N to 1 transfer rule.

ColumnType Type
Sets or returns column type. Read/write.

String TypeName
Returns name of the column type. Read only.

Boolean Unique
Sets or returns unique flag of the linked column. Read/write.

**Methods**

Flatten()
Converts active column, rank column or column brought by relation to native column.

Boolean IsActive()
Returns True for active columns and False for other columns.

Boolean IsForeign()
Returns True for columns mapped from another table and False for other columns.

Boolean IsIntrinsic()
Returns True for intrinsic columns and False for other columns.

Boolean IsKey()
Returns True for linked columns that are primary keys and False for other columns.

Boolean IsLinked()
Returns True for linked columns and False for other columns.
Boolean IsLookup()
    Returns True for lookup columns and False for other columns.

Boolean IsNative()
    Returns True for native columns and False for other columns.

Boolean IsQuery()
    Returns True for columns created by the query and False for other columns.

Boolean IsRank()
    Returns True for rank columns and False for other columns.

Boolean IsTypeBinary()
    Returns True for binary columns and False for other columns.

Boolean IsTypeBoolean()
    Returns True for boolean columns and False for other columns.

Boolean IsTypeCoordSys()
    Returns True for coordinate system columns and False for other columns.

Boolean IsTypeCurrency()
    Returns True for currency columns and False for other columns.

Boolean IsTypeGeometry()
    Returns True for geometry columns and False for other columns.

Boolean IsTypeLatitude()
    Returns True for latitude columns and False for other columns.

Boolean IsTypeLongitude()
    Returns True for longitude columns and False for other columns.

Boolean IsTypeNumeric()
    Returns True for numeric columns and False for other columns.

Boolean IsTypePercent()
    Returns True for percentage columns and False for other columns.

Boolean IsTypeString()
    Returns True for text columns and False for other columns.

Boolean IsTypeSymbol()
    Returns True for character columns and False for other columns.

Boolean IsTypeTime()
    Returns True for date and time columns and False for other columns.

Boolean IsTypeUnsigned()
    Returns True for unsigned columns and False for other columns.

Boolean IsTypeUrl()
Returns True for URL columns and False for other columns.

Boolean IsTypeValid()
Returns True if column type is valid and False otherwise.

Boolean IsTypeVector()
Returns True for vector columns and False otherwise.

Paste()
Pastes the contents of the Clipboard into the column.

Recompute()
Recomputes active column.

Obtained From

Column (OriginColumn)
ColumnSet (Item, LastAdded, NewColumn)
DSSAtom (Column)
Format (Column)
Relation (Column, TargetColumn)
Table (Identity)
TableWindow (ActiveColumn)

See Also

Scripting Reference
**ColumnFormat Object**
Formatting options for table column.

**Properties**

ColumnAlign Align
Sets or returns column alignment. Read/write.

Application Application
Returns application. Read only.

Number Decimals
Sets or returns number of decimal digits. Read/write.

Number Indent
Sets or returns indentation in pixels. Read/write.

ColumnFormatNeg NegativeStyle
Sets or returns formatting options for negative values. Read/write.

Any Parent
Returns parent object. Read only.

ColumnFormatPos PositiveStyle
Sets or returns formatting options for positive values. Read/write.

String Style
Sets or returns format style. Read/write.

Boolean Thousands
Sets or returns thousand formatting option. Read/write.

Number Width
Sets or returns column width in pixels. Read/write.

**Obtained From**

Column (Format)

**See Also**

Scripting Reference
**ColumnLookupSet Object**
Set of values for lookup columns.

**Properties**

Applycation Application
Returns application. Read only.

Number Count
Returns number of lookup values in set. Read only.

LookupValue Item(Number index)
Returns lookup value with given index. Read only.

LookupValue LastAdded
Returns last added lookup value. Read only.

Any Parent
Returns parent object. Read only.

**Methods**

Add(LookupValue value)
Appends lookup value to lookup set.

Number ItemByCaption(String caption)
Returns index of value with given caption or -1.

LookupValue NewLookupValue()
Returns lookup value that can later be added to lookup set.

Remove(Number item)
Removes lookup value with given index from lookup set.

**Obtained From**

Application (NewColumnLookupSet)
Column (LookupValues)

**See Also**

Scripting Reference
**ColumnSet Object**
Set of table columns.

**Properties**

Application Application
Returns application. Read only.

Number Count
Returns number of columns in set. Read only.

Column Item(Number index)
Column Item(String name)
Returns column with given index or name. Read only.

Column LastAdded
Returns last added column if any. Read only.

Component OwnerComponent
Returns table component containing this set of columns if any. Read only.

Any Parent
Returns parent object. Read only.

**Methods**

Add(Column column)
Appends column to column set.

Number ItemByID(Number id)
Returns index of column with given ID or -1.

Number ItemByName(String name)
Returns index of column with given name or -1.

Column NewColumn()
Creates new column that can later be added into column set.

Remove(Number index)
Removes column with given index from column set.

RemoveAll()
Removes all columns.

**Obtained From**

Application (NewColumnSet)
Chart (Groups, Series)
LayoutEntry (ColumnSet)
Table (ColumnSet)

**See Also**
Scripting Reference
Comments Object
Comments component.

Properties

Application Application
Returns application. Read only.

Boolean Cached
Sets or returns caching option for shared component in Enterprise Edition. Read/write.

String Description
Sets or returns component description. Read/write.

Document Document
Returns owner document. Read only.

Folder Folder
Sets or returns containing folder. Read/write.

Number ID
Returns unique component ID. Read only.

Boolean IsCheckedOut
Checks if component is shared and checked out in Enterprise Edition. Read only.

Boolean IsCheckedOutRemotely
Checks if component is shared and checked out by another user in Enterprise Edition. Read only.

Boolean IsModified
Checks if component is modified since last save in Enterprise Edition. Read only.

Boolean IsOutdated
Checks if component is shared and there is a later version of the component on the Enterprise server in Enterprise Edition. Read only.

Boolean IsReadOnly
Checks if component is readonly (eg, shared and not checked out) in Enterprise Edition. Read only.

Boolean IsShared
Checks if component is shared on an Enterprise server in Enterprise Edition. Read only.

Boolean IsWritable
Checks if component is writable in Enterprise Edition. Read only.

Number Length
Returns number of characters in comments text. Read only.

String Name
Sets or returns component name. Read/write.

String Note
Sets or returns notes associated with the component. Read/write.
ComponentSet OwnedComponentSet
Returns set of components bound to this component. Read only.

ComponentSet OwnedLayoutSet
Returns set of layouts bound to this component. Read only.

Component Owner
Returns owner component if any. Read only.

Any Parent
Returns parent object. Read only.

Number RemoteVersion
Returns version of shared component available on the Enterprise server in Enterprise Edition. Read only.

String Server
Returns server identification string for shared component in Enterprise Edition. Read only.

String ShareStatus
Returns textual report on component sharing status in Enterprise Edition. Read only.

String Text
Sets or returns comments text. Read/write.

ComponentType Type
Returns component type. Read only.

String TypeName
Returns textual representation of the component type. Read only.

Number Version
Returns version of shared component within the local project in Enterprise Edition. Read only.

Methods

AddText(String text)
Appends text to the end of the component.

CheckIn()
Checks in changes made to a shared component to the Enterprise server in Enterprise Edition.

CheckOut()
Checks out a shared component for editing in Enterprise Edition.

Clear()
Clears component.

Copy()
Copies component into the Clipboard.

Cut()
Copies component into the Clipboard and then clears it.
GetLatestVersion()
Gets the latest version of a shared component from the Enterprise server in Enterprise Edition.

Boolean IsEmpty(Boolean ignoreWhiteSpaces)
Returns True if component text is empty and False otherwise.

Open()
Opens component.

OpenInNewWindow()
Opens component in new window.

Paste(Boolean replace)
Pastes the contents of the Clipboard into the component replacing previous data or adding to it. Calling the method without parameters replaces data.

Print(Boolean useDialog)
Prints component.

UndoCheckOut()
Undoes changes made to a shared component and gets the latest version of the component from the Enterprise server in Enterprise Edition.

Obtained From
Document (NewComments)

See Also
Scripting Reference
**Component Object**
Generic component.

**Properties**

Application Application
Returns application. Read only.

Boolean Cached
Sets or returns caching option for shared component in Enterprise Edition. Read/write.

String Description
Sets or returns component description. Read/write.

Document Document
Returns owner document. Read only.

Folder Folder
Sets or returns containing folder. Read/write.

Number ID
Returns unique component ID. Read only.

Boolean IsCheckedOut
Checks if component is shared and checked out in Enterprise Edition. Read only.

Boolean IsCheckedOutRemotely
Checks if component is shared and checked out by another user in Enterprise Edition. Read only.

Boolean IsModified
Checks if component is modified since last save in Enterprise Edition. Read only.

Boolean IsOutdated
Checks if component is shared and there is a later version of the component on the Enterprise server in Enterprise Edition. Read only.

Boolean IsReadOnly
Checks if component is readonly (eg, shared and not checked out) in Enterprise Edition. Read only.

Boolean IsShared
Checks if component is shared on an Enterprise server in Enterprise Edition. Read only.

Boolean IsWritable
Checks if component is writable in Enterprise Edition. Read only.

String Name
Sets or returns component name. Read/write.

String Note
Sets or returns notes associated with the component. Read/write.

ComponentSet OwnedComponentSet
Returns set of components bound to this component. Read only.
Component Owner
Returns owner component if any. Read only.

Any Parent
Returns parent object. Read only.

Number RemoteVersion
Returns version of shared component available on the Enterprise server in Enterprise Edition. Read only.

String Server
Returns server identification string for shared component in Enterprise Edition. Read only.

String ShareStatus
Returns textual report on component sharing status in Enterprise Edition. Read only.

ComponentType Type
Returns component type. Read only.

String TypeName
Returns textual representation of the component type. Read only.

Number Version
Returns version of shared component within the local project in Enterprise Edition. Read only.

Methods

CheckIn()
Checks in changes made to a shared component to the Enterprise server in Enterprise Edition.

CheckOut()
Checks out a shared component for editing in Enterprise Edition.

Get()
Gets the latest version of a shared component from the Enterprise server in Enterprise Edition.

Open()
Opens component.

OpenInNewWindow()
Opens component in new window.

Print(Boolean useDialog)
Prints component.

Run()
Runs component.

UndoCheckOut()
Undoes changes made to a shared component and gets the latest version of the component from the Enterprise server in Enterprise Edition.

Obtained From
Chart (Owner)
Column (OwnerComponent)
ColumnSet (OwnerComponent)
Comments (Owner)
Component (Owner)
ComponentSet (Item)
Drawing (Owner)
DrawingWindow (ActiveComponent, Component)
Folder (Owner)
Form (Owner)
Format (Component)
Image (Owner)
Label (OwnerComponent)
Labels (Owner)
LabelSet (OwnerComponent)
LabelWindow (ActiveComponent, Component)
Layer (Component, OwnerComponent)
LayerSet (OwnerComponent)
Layout (Owner)
LayoutEntry (Component, OwnerComponent)
LayoutEntrySet (OwnerComponent)
LayoutWindow (ActiveComponent, Component)
Object (OwnerComponent)
ObjectSet (OwnerComponent)
Map (Owner)
MapWindow (ActiveComponent, Component)
Palette (Owner)
Pixel (OwnerComponent)
PixelSet (OwnerComponent)
Query (Owner)
Record (OwnerComponent)
RecordSet (OwnerComponent)
Script (Owner)
Surface (Owner)
Table (Owner)
TableWindow (ActiveComponent, Component)
Terrain (Owner)
View (Component)
Window (ActiveComponent, Component)

See Also

Scripting Reference
ComponentSet Object
Set of components.

Properties

Application Application
Returns application. Read only.

Number Count
Returns number of components in set. Read only.

Document Document
Returns owner document. Read only.

Component Item(Number index)
Component Item(String name)
Returns component with given index or name. Read only.

Any Parent
Returns parent object. Read only.

Methods

Add(Component component)
Appends component to component set.

Number ItemByID(Number id)
Returns index of component with given ID or -1.

Number ItemByName(String name)
Returns index of component with given name or -1.

Remove(Component component)
Remove(Number index)
Remove(String name)
Removes given component or component with given index or name.

RemoveAll()
Removes all components.

Obtained From

Document (ComponentSet, NewComponentSet)
Drawing (OwnedLabelSet)
Folder (Children)
Terrain (Overlay)

See Also

Scripting Reference
ControlPoint Object
Control point.

Properties

Application Application
Returns application. Read only.

Point Location
Sets or returns control point location. Read/write.

String Name
Sets or returns control point name. Read/write.

Any Parent
Returns parent object. Read only.

Obtained From

Application (NewControlPoint)
ControlPointSet (Item, LastAdded)

See Also

Scripting Reference
**ControlPointSet Object**
Set of control points.

**Properties**

Application Application
Returns application. Read only.

Number Count
Returns number of control points in set. Read only.

ControlPoint Item(Number index)
ControlPoint Item(String name)
Returns control point with given index or name. Read only.

ControlPoint LastAdded
Returns last added control point. Read only.

Any Parent
Returns parent object. Read only.

**Methods**

Add(ControlPoint point)
Appends control point to control point set.

Number ItemByName(String name)
Returns index of control point with given name or -1.

Remove(Number index)
Removes control point with given index.

RemoveAll()
Removes all control points.

**Obtained From**

Drawing (ControlPointSet)
Image (ControlPointSet)
Labels (ControlPointSet)
Map (ControlPointSet)
Surface (ControlPointSet)
Zones (ControlPointSet)

**See Also**

Scripting Reference
ConverterItemSet Object
Set of converter items such as fields or coverages.

Properties

Application Application
Returns application. Read only.

Number Count
Returns number of items in set. Read only.

String Item(Number index)
Returns item with given index. Read only.

String LastAdded
Returns last added item if any. Read only.

Any Parent
Returns parent object. Read only.

Methods

Add(String name)
Appends item to item set.

Remove(Number index)
Removes item with given index from item set.

Remove(String item)
Removes given item from item set.

RemoveAll()
Removes all items from item set.

Obtained From

ExportCsv (Columns)
ExportDb (Columns)
ExportDbf (Columns)
ExportDxf (Columns)
ExportE00 (Columns)
ExportHtml (Columns)
ExportMdb (Columns)
ExportMfd (Columns)
ExportMif (Columns)
ExportSdts (Columns)
ExportShp (Columns)
ExportWk (Columns)
ExportXls (Columns)
ImportAvhhr (Modules)
ImportBil (Modules)
ImportBna (Columns)
ImportCadrgCib (Modules)
ImportCeosSeaWifs (Modules)
ImportCsv (Columns)
ImportDb (Columns)
ImportDbf (Columns)
ImportDlg (Columns)
ImportE00 (Columns)
ImportEnvi (Modules)
ImportErdasImagine (Modules)
ImportErs (Modules)
ImportGdf (Columns)
ImportGeoSpot (Modules)
ImportGml (Updates)
ImportHdf (Modules)
ImportHdfEos (Modules)
ImportHdfSeaWifs (Modules)
ImportHtml (Tables)
ImportLas (Modules)
ImportLulc (Modules)
ImportMapBase (Columns)
ImportMdb (Tables)
ImportMfd (Columns)
ImportMif (Columns)
ImportNtad (Columns)
ImportNtf (Columns)
ImportOleDb (Tables)
ImportPix (Modules)
ImportSdts (Columns)
ImportShp (Columns)
ImportSpot (Modules)
ImportTab (Columns)
ImportTaif (Modules)
ImportTiger (Columns)
ImportUdl (Tables)
ImportVmap (Modules)
ImportWk (Tables)
ImportXls (Tables)

See Also

Scripting Reference
**ConverterProperty Object**
Converter property.

**Properties**

Application Application
Returns application. Read only.

String Name
Sets or returns property name. Read/write.

Any Parent
Returns parent object. Read only.

String Value
Sets or returns property value. Read/write.

**Obtained From**

Application (NewConverterProperty)
ConverterPropertySet (Item, LastAdded)

**See Also**

Scripting Reference
ConverterPropertySet Object
Set of converter properties.

Properties

Application Application
Returns application. Read only.

Number Count
Returns number of properties in set. Read only.

ConverterProperty Item(Number index)
Returns property with given index. Read only.

ConverterProperty LastAdded
Returns last added property if any. Read only.

Any Parent
Returns parent object. Read only.

Methods

Add(ConverterProperty property)
Appends property to property set.

Remove(Number index)
Removes property with given index from property set.

Remove(String name)
Removes property with given name from property set.

RemoveAll()
Removes all properties from property set.

Obtained From

Export (PropertySet)
Import (PropertySet)

See Also

Scripting Reference
**CoordinateConverter Object**
Coordinate system converter.

**Methods**

Convert(Branch branch, Variant status)
Converts set of points in a branch and writes conversion status for each point.

Convert(BranchSet branchSet, Variant status)
Converts set of points in a set of branches and writes conversion status for each point.

Convert(Geom geom, Variant status)
Converts geometric entity and writes conversion status for each point.

Convert(GeomSet geomSet, Variant status)
Converts set of geometric entities and writes conversion status for each point.

Convert(Point pointSet, Variant status)
Converts point and writes conversion status.

Convert(PointSet pointSet, Variant status)
Converts set of points and writes conversion status for each point.

Boolean ConvertsData()
Returns True if source and target coordinate systems are not the same and False otherwise.

Prepare(CoordinateSystem source, CoordinateSystem target)
Prepares coordinate converter with source and target coordinate systems.

**Obtained From**

Application (NewCoordinateConverter)

**See Also**

Scripting Reference
CoordinateSystem Object
Coordinate system.

For conversions between two different coordinate systems use CoordinateConverter Object.

Properties

Application Application
Returns application. Read only.

String Category
Returns coordinate system category. Read only.

Datum Datum
Sets or returns datum. Read/write.

String Name
Returns coordinate system name. Read only.

CoordinateSystemParameterSet ParameterSet
Returns set of coordinate system parameters. Read only

Any Parent
Returns parent object. Read only.

String Preset
Returns preset used to create coordinate system. Read only.

Unit Unit
Sets or returns unit. Read/write.

Methods

Point ConvertForward(Point point)
Converts point from latitude/longitude to coordinate system.

Point ConvertInverse(Point point)
Converts point from coordinate system to latitude/longitude.

Copy(CoordinateSystem coordinateSystem)
Copies given coordinate system.

Boolean Edit()
Edits coordinate system in a dialog. Returns True if editing has been accepted and False otherwise.

Boolean IsEqualTo(CoordinateSystem system, Boolean compareShiftsScales, Boolean compareDatums)
Compares coordinate system to another coordinate system. Returns True if coordinate systems are equal to each other and False otherwise. Two last parameters are optional (both are True by default).

Boolean IsLatLon()
Returns True if coordinate system transforms locations to latitude/longitude and False otherwise.

Load(String name)
Loads coordinate system using given name.

LoadFrom(String xml)
Loads coordinate system from given XML string.

LoadFromBinary(Object binary)
 Loads coordinate system from given binary data.

LoadFromFile(String path)
Loads coordinate system from given XML file.

Object ToBinary()
Creates a binary representation of a coordinate system.

SaveToFile(String path)
Saves coordinate system to given XML file.

String ToTextGSR(Number width, Number height)
Saves coordinate system to GSR string. The last two parameters are the width and height of an image or surface which has the coordinate system, and are optional.

String ToTextPRJ()
Saves coordinate system to PRJ string.

String ToTextWKT()
Saves coordinate system to WKT string (same as PRJ string).

String ToXML()
Saves coordinate system to XML string.

Obtained From

Application (NewCoordinateSystem, DefaultCoordinateSystem, DefaultCoordinateSystemLatLon)
Drawing (CoordinateSystem)
Image (CoordinateSystem)
Labels (CoordinateSystem)
Map (CoordinateSystem)
Surface (CoordinateSystem)
Theme (CoordinateSystem)
Zones (CoordinateSystem)

See Also

Scripting Reference
CoordinateSystemParameter Object
Coordinate system parameter. Parameters with maximum value lower than minimum value can have any numeric value. Parameters with maximum value higher than minimum value can have values between those two.

Properties

**Application**
Returns application. Read only.

**Number MaxValue**
Returns maximum possible value. Read only.

**Number MinValue**
Returns minimum possible value. Read only.

**String Name**
Returns parameter name. Read only.

**Any Parent**
Returns parent object. Read only.

**CoordinateSystemParameterType Type**
Returns parameter type. Read only.

**String TypeName**
Returns textual representation of parameter type. Read only.

**Number Value**
Sets or returns parameter value. Read/write.

**Obtained From**

CoordinateSystemParameterSet (Item)

**See Also**

Scripting Reference
CoordinateSystemParameterSet Object
Set of coordinate system parameters.

Properties

Application Application
Returns application. Read only.

Number Count
Returns number of parameters in set. Read only.

CoordinateSystemParameter Item(Number index)
CoordinateSystemParameter Item(String name)
Returns parameter with given index or name. Read only.

Any Parent
Returns parent object. Read only.

Methods

Number ItemByName(String name)
Returns index of parameter with given name or -1.

Obtained From

CoordinateSystem (Parameters)

See Also

Scripting Reference
CoordinateSystemSet Object
Set of coordinate systems.

Properties

Application Application
Returns application. Read only.

Number Count
Returns number of coordinate systems in set. Read only.

CoordinateSystem Item(Number index)
CoordinateSystem Item(String name)
Returns coordinate system with given index or name. Read only.

Any Parent
Returns parent object. Read only.

Methods

Number ItemByName(String name)
Returns index of coordinate system with given name or -1.

Obtained From

Application (CoordinateSystemSet)

See Also

Scripting Reference
**DataSource Object**

A data source.

**Properties**

- **Application**
  - **Application**
    - Returns application. Read only.

- **ConnectionString**
  - **String**
    - Sets or returns connection string. Read/write.

- **ConnectionType**
  - **String**

- **Parent**
  - **Any**
    - Returns parent object. Read only.

**Methods**

- **ExportDrawing(Drawing drawing, String name, String type, Number srid, Boolean createIndex, Boolean createSequenceTriggers, DataSourceColumnSet columns)**
  - Exports a drawing using the specified name and drawing type. If the SRID parameter is set to -1, the system uses the coordinate system that best matches the coordinate system of the exported component. All parameters except the first three are optional.

- **ExportImage(Image image, String name, String type, Number srid, Boolean createPyramids, String compression)**
  - Exports an image using the specified name and image type. If the SRID parameter is set to -1, the system uses the coordinate system that best matches the coordinate system of the exported component. All parameters except the first three are optional.

- **ExportSurface(Surface surface, String name, String type, Number srid, Boolean createPyramids, String compression)**
  - Exports a surface using the specified name and surface type. If the SRID parameter is set to -1, the system uses the coordinate system that best matches the coordinate system of the exported component. All parameters except the first three are optional.

- **ExportTable(Table table, String name, DataSourceColumnSet columns)**
  - Exports a table using the specified name. The last parameter is optional.

- **ImportDrawing(String drawing, DataSourceColumnSet columns)**
  - Imports a native or Manifold drawing with specified name. The last parameter is optional.

- **ImportDrawingFromTableCoordinates(String table, String columnX, String columnY, Boolean latLon, String columnLineID, String columnCoordinateOrder, DataSourceColumnSet columns)**
  - Imports a drawing from a table with coordinate columns. All parameters except the first three are optional.

- **ImportDrawingFromTableGeometry(String table, String columnGeometry, ColumnType geometryType, String columnVersion, DataSourceColumnSet columns)**
  - Imports a drawing from a table with a geometry column. All parameters except the first three are optional.

- **ImportImage(String image)**
  - Imports a native or Manifold image with specified name.
ImportImageFromTableChannels(String table, Number channels, String columnX, String columnY, String columnRed, String columnGreen, String columnBlue, String columnAlpha)
Imports an image from a table with channel columns. The number of channels can be 1, 3 or 4. All parameters except the first five are optional.

ImportImageFromTableColor(String table, Number channels, String columnX, String columnY, String columnColor)
Imports an image from a table with a color column. The number of channels can be 1, 3 or 4.

ImportSurfaceFromImageChannel(String image, Number channel)
Imports a surface from a channel of a native or Manifold image with specified name. If the channel index is set to -1, imports the heights channel. The channel index parameter is optional.

ImportSurfaceFromTableValue(String table, ColumnType resultType, String columnX, String columnY, String columnValue)
Imports a surface from a table with a value column.

ImportTable(String table, DataSourceColumnSet columns)
Imports a table with specified name. The last parameter is optional.

LinkDrawing(String drawing, DataSourceColumnSet columns)
Links a native or Manifold drawing with specified name. The last parameter is optional.

LinkDrawingFromTableCoordinates(String table, String columnX, String columnY, Boolean latLon, String columnLineID, String columnCoordinateOrder, DataSourceColumnSet columns)
Links a drawing from a table with coordinate columns. All parameters except the first three are optional.

LinkDrawingFromTableGeometry(String table, String columnGeometry, ColumnType geometryType, String columnVersion, DataSourceColumnSet columns)
Links a drawing from a table with a geometry column. All parameters except the first three are optional.

LinkImage(String image)
Links a native or Manifold image with specified name.

LinkImageFromTableChannels(String table, Number channels, String columnX, String columnY, String columnRed, String columnGreen, String columnBlue, String columnAlpha)
Links an image from a table with channel columns. The number of channels can be 1, 3 or 4. All parameters except the first five are optional.

LinkImageFromTableColor(String table, Number channels, String columnX, String columnY, String columnColor)
Links an image from a table with a color column. The number of channels can be 1, 3 or 4.

LinkSurfaceFromTableValue(String table, ColumnType resultType, String columnX, String columnY, String columnValue)
Links a surface from a table with a value column.

LinkTable(String table, DataSourceColumnSet columns)
Links a table with specified name. The last parameter is optional.

Boolean Probe()
Tries to connect to the data source. Returns True if the data source can be reached, and False otherwise.

 Obtained From

Document (NewDataSource)
See Also

Scripting Reference
**DataSourceColumnSet Object**

A set of column names to export or import.

**Properties**

**Application Application**

Returns application. Read only.

**Number Count**

Returns number of column names in set. Read only.

**String Item(Number index)**

Returns column name with given index. Read only.

**String LastAdded**

Returns last added column name. Read only.

**Any Parent**

Returns parent object. Read only.

**Methods**

**Add(String column)**

Adds a column name to a set.

**Remove(Number index)**

Removes a column name with given index.

**RemoveAll()**

Removes all column names.

**Obtained From**

Application (NewDataSourceColumnSet)

**See Also**

Scripting Reference
Datum Object
Datum.

Properties

Application Application
Returns application. Read only.

Ellipsoid Ellipsoid
Returns reference ellipsoid. Read/write

String Name
Returns datum name. Read only.

Any Parent
Returns parent object. Read only.

Number ScaleAdjustment
Sets or returns scale adjustment, in units per million. Read/write.

Number XRotation
Sets or returns rotation along the X axis, in degrees. Read/write.

Number XShift
Sets or returns offset along the X axis, in meters. Read/write.

Number YRotation
Sets or returns offset along the Y axis, in meters. Read/write.

Number YShift
Returns center shift by Y axis. Read/write.

Number ZRotation
Sets or returns rotation along the Z axis, in degrees. Read/write.

Number ZShift
Sets or returns offset along the Z axis, in meters. Read/write.

Methods

Load(String name)
Loads datum using given name.

 Obtained From

 Application (NewDatum, DefaultDatum)
 CoordinateSystem (Datum)
 DatumSet (Item)

See Also

Scripting Reference
**DatumSet Object**
Set of datums.

**Properties**

- **Application Application**
  Returns application. Read only.

- **Number Count**
  Returns number of datums in set. Read only.

- **Datum Item(Number index)**
- **Datum Item(String name)**
  Returns datum with given index or name. Read only.

- **Any Parent**
  Returns parent object. Read only.

**Methods**

- **Number ItemByName(String name)**
  Returns index of datum with given name or -1.

**Obtained From**

- **Application (DatumSet)**

**See Also**

- **Scripting Reference**
**Document Object**  
Manages components and document-wide properties.

**Properties**

**Boolean Active**  
Returns True if the document is the active document displayed by the Manifold user interface, and False otherwise. Read only.

**Application Application**  
Returns application. Read only.

**String Author**  
Sets or returns document author. Read/write.

**Boolean BatchUpdates**  
Sets or returns batch update mode. When batch mode is on, changes made by script are not immediately propagated to the mapfile, which is faster but can make script objects report outdated information. The recommended technique is to turn BatchUpdates on when changing large bulks of data and turn it back off when all necessary changes have been done. Read/write.

**String Comments**  
Sets or returns comments associated with document. Read/write.

**ComponentSet ComponentSet**  
Returns set of document components. Read only.

**String FullName**  
Returns fully qualified filename of a document. Returns empty string for newly created documents. Read only.

**String Keywords**  
Sets or returns list of keywords associated with document. Read/write.

**String Name**  
Returns document name. Read only.

**Any Parent**  
Returns parent object. Read only.

**String Path**  
Returns path to the document file. Read only.

**Boolean ReadOnly**  
Returns True if document is opened in read only mode and False otherwise. Read only.

**Boolean Saved**  
Returns True if document has been saved after the latest modifications and False otherwise. Read only.

**String Subject**  
Sets or returns subject associated with document. Read/write.

**String Title**  
Sets or returns document title. Read/write.
Methods

CheckIn(ComponentSet components)
Checks in changes made to shared components to the Enterprise server or servers in Enterprise Edition.

CheckInAll()
Checks in changes made to all shared components in the document to the Enterprise server or servers in Enterprise Edition.

CheckOut(ComponentSet components)
Checks out shared components for editing in Enterprise Edition.

CheckOutAll()
Checks out all shared components in the document for editing in Enterprise Edition.

Close(Boolean saveChanges)
Closes document saving or discarding changes. If argument is omitted, the changes are saved. If document contains running script the method fails.

Clear()
Removes all components excluding running script component and any components it depends on (tables, drawings, folders).

GetLatestVersion(ComponentSet components)
Gets the latest version of shared components from the Enterprise server or servers in Enterprise Edition.

GetLatestVersionAll()
Gets the latest version of all shared components in the document from the Enterprise server or servers in Enterprise Edition.

Analyzer NewAnalyzer()
Creates new analyzer object.

Chart NewChart(String name, Table table, Boolean exactName)
Creates new chart component for given table. Exact name parameter is optional. When it is set to False or omitted, the system adjusts the name so that there are no collisions with existing components.

Comments NewComments(String name, Boolean exactName)
Creates new comments component. Exact name parameter is optional. When it is set to False or omitted, the system adjusts the name so that there are no collisions with existing components.

ComponentSet NewComponentSet()
Creates new component set object for later use in NewMap and similar methods.

DataSource NewDataSource()
Creates a new data source for exporting or importing data.

Drawing NewDrawing(String name, CoordinateSystem system, Boolean exactName)
Creates new drawing component. Coordinate system and exact name parameters are optional. When exact name is set to False or omitted, the system adjusts the name so that there are no collisions with existing components.

Elevation NewElevation(String name, Profile profile, Boolean exactName)
Creates new elevation component for given profile. Exact name parameter is optional. When it is set to False or omitted, the system adjusts the name so that there are no collisions with existing components.

Export NewExport(String name)
Creates new exporter using exporter name.

Folder NewFolder(String name, Boolean exactName)
Creates new folder component. Exact name parameter is optional. When it is set to False or omitted, the system adjusts the name so that there are no collisions with existing components.

Form NewForm(String name, Boolean exactName)
Creates new form component. Exact name parameter is optional. When it is set to False or omitted, the system adjusts the name so that there are no collisions with existing components.

Image NewImage(String name, Number width, Number height, ImageType type, CoordinateSystem system, Boolean exactName)
Creates new image component. Width, height, type, coordinate system and exact name parameters are optional. When exact name is set to False or omitted, the system adjusts the name so that there are no collisions with existing components.

Import NewImport(String name)
Creates new importer using importer name.

Labels NewLabels(String name, CoordinateSystem system, Boolean exactName)
Creates new labels component. Coordinate system and exact name parameters are optional. When exact name is set to False or omitted, the system adjusts the name so that there are no collisions with existing components.

Labels NewLabels(String name, Drawing drawing, Boolean exactName, Boolean populate)
Creates new labels component linked to given drawing. The last two parameters are optional. When the exact name parameter is set to False or omitted, the system adjusts the name so that there are no collisions with existing components. When the populate parameter is set to True, the system automatically populates the labels component with a label for each drawing object.

Layer NewLayer(Component component, Number opacity)
Creates new layer with given component. Opacity parameter is optional and can vary from 100 (opaque) to 0 (transparent).

Layout NewLayout(String name, Component component, Boolean exactName, String template)
Creates new layout component. Component, exact name and template parameters are optional. When exact name is set to False or omitted, the system adjusts the name so that there are no collisions with existing components.

Map NewMap(String name, Component layer, CoordinateSystem system, Boolean exactName)
Map NewMap(String name, ComponentSet layers, CoordinateSystem system, Boolean exactName)
Creates new map component. Coordinate system and exact name parameters are optional.

ObjectSet NewObjectSet(Object content)
ObjectSet NewObjectSet(ObjectSet content)
Creates new static object set with given content. Content parameter is optional.

Palette NewPalette(String name, Boolean exactName)
Creates new palette component. Exact name parameter is optional. When it is set to False or omitted, the system adjusts the name so that there are no collisions with existing components.

Profile NewProfile(String name, Surface surface, Branch metric, Boolean exactName)
Profile NewProfile(String name, Surface surface, BranchSet metric, Boolean exactName)
Profile NewProfile(String name, Surface surface, Geom metric, Boolean exactName)
Profile NewProfile(String name, Surface surface, PointSet metric, Boolean exactName)

Creates new profile component for given surface. Metric parameter is optional. Exact name parameter is optional. When it is set to False or omitted, the system adjusts the name so that there are no collisions with existing components.

Query NewQuery(String name, Boolean exactName)

Creates new query component. Exact name parameter is optional. When it is set to False or omitted, the system adjusts the name so that there are no collisions with existing components.

Script NewScript(String name, Boolean exactName)

Creates new script component. Exact name parameter is optional. When it is set to False or omitted, the system adjusts the name so that there are no collisions with existing components.

Surface NewSurface(String name, Number width, Number height, ValueType type, CoordinateSystem system, Boolean exactName)

Creates new surface component. Width, height, type, coordinate system and exact name parameters are optional. When exact name is set to False or omitted, the system adjusts the name so that there are no collisions with existing components.

Table NewTable(String name, ColumnSet columns, Boolean exactName)

Creates new table component. Column set and exact name parameters are optional. When exact name is set to False or omitted, the system adjusts the name so that there are no collisions with existing components.

Terrain NewTerrain(String name, Surface surface, Boolean exactName)

Creates new terrain component for given surface. Exact name parameter is optional. When it is set to False or omitted, the system adjusts the name so that there are no collisions with existing components.

Theme NewTheme(String name, Component parent, Boolean exactName)

Creates new theme component linked to given drawing or theme. Exact name parameter is optional. When it is set to False or omitted, the system adjusts the name so that there are no collisions with existing components.

Zones NewZones(String name, CoordinateSystem system, Boolean exactName)

Creates new zones component. Coordinate system and exact name parameters are optional. When it is set to False or omitted, the system adjusts the name so that there are no collisions with existing components.

Open(String fileName, Boolean saveChanges, Boolean readOnly)

Opens new document saving or discarding any pending changes.

Paste()

Pastes the contents of the Clipboard as new component.

PasteAs(ComponentType type)

Pastes the contents of the Clipboard as new component of the desired type.

RefreshAllLinked()

Refreshes all linked components.

RevertToSaved()

Reverts documents to last saved state if any.

Save()

Saves document.

SaveAs(String fileName)

Saves document with a new name.
UndoCheckOut(ComponentSet components)
Unused changes made to shared components and gets the latest version of the components from the Enterprise
server or servers in Enterprise Edition.

UndoCheckOutAll()
Undoes changes made to all shared components in the document and gets the latest version of the components
from the Enterprise server or servers in Enterprise Edition.

**Tech Note**

Prior to 6.50 SP1, the BatchUpdates property applied only to the active document (so, there was no method to
increase performance by batching updates in an external application or on a web site) and was global in that all
instances of the Document object referring to the MAP file opened in the UI were sharing the value of the
property. As of 6.50 SP1, the BatchUpdates property works for all documents and is scoped to a particular
document object.

That is,

Application.ActiveDocument.BatchUpdates = Xxx

...sets the property in a temporary instance of the Document object, which does not have any effect.

However,

Set doc = Application.ActiveDocument
doc.BatchUpdates = True
...
... * use data obtained from the "doc" object ...

...batches updates for all write operations in all objects retrieved from the "doc" object.

The new policy is both faster and more flexible.

**Obtained From**

Application (NewDocument, ActiveDocument)
Chart (Document)
Comments (Document)
Component (Document)
ComponentSet (Document)
DocumentSet (Item)
Drawing (Document)
Elevation (Document)
Folder (Document)
Form (Document)
Image (Document)
Labels (Document)
Layout (Document)
Map (Document)
Palette (Document)
Profile (Document)
Query (Document)
Script (Document)
Surface (Document)
See Also

Scripting Reference
**DocumentSet Object**

Set of documents.

**Properties**

Application Application
Returns application. Read only.

Number Count
Returns number of documents in set. Read only.

Document Item(Number index)
Returns document with given index. Read only.

Any Parent
Returns parent object. Read only.

**Methods**

Close(Boolean saveChanges)
Closes all documents within set except documents containing running scripts. If argument is omitted, the changes are saved.

Open(String fileName, Boolean readOnly)
Opens new document.

**Obtained From**

Application (DocumentSet)

**See Also**

Scripting Reference
**Drawing Object**

Drawing component.

**Properties**

Application Application
Returns application. Read only.

Format AreaBackground
Returns background color used for areas. Read only.

Format AreaBorderBackground
Returns border background color used for areas. Read only.

Format AreaBorderForeground
Returns border foreground color used for areas. Read only.

Format AreaBorderSize
Returns border size used for areas. Read only.

Format AreaBorderStyle
Returns border style used for areas. Read only.

Format AreaForeground
Returns foreground color used for areas. Read only.

Format AreaSize
Returns size used for areas. Read only.

Format AreaStyle
Returns style used for areas. Read only.

Color BackgroundColor
Sets or returns background color. Read / write.

Rect Box
Returns bounding box of all objects. Read only.

Boolean Cached
Sets or returns caching option for shared component in Enterprise Edition. Read/write.

Boolean CanAddObjects
Returns True if the drawing supports adding new objects and False otherwise. Read only.

Boolean CanAlterObjects
Returns True if the drawing supports altering object metric and False otherwise. Read only.

Boolean CanHaveLinkArea
Returns True if the linked drawing supports setting the area of interest and False otherwise. Read only.

Boolean CanRemoveObjects
Returns True if the drawing supports removing objects and False otherwise. Read only.
ControlPointSet ControlPointSet
Returns set of control points. Read only.

CoordinateSystem CoordinateSystem
Sets or returns coordinate system. Read/write.

Boolean CoordinateSystemVerified
Checks whether or not the component coordinate system has been verified by the user, or alters the verification state. Read/write.

View DefaultView
Returns default view. Read only.

String Description
Sets or returns component description. Read/write.

Document Document
Returns owner document. Read only.

Number Epsilon
Sets or returns location precision. Read/write.

Folder Folder
Sets or returns containing folder. Read/write.

Number GeometryVersion
Returns a numeric value representing the version of the geometry inside the drawing. Adding or removing drawing objects or altering their metric advances the version number. Read only.

Boolean HasLinkArea
Returns True if the linked drawing uses the area of interest and False otherwise. Read only.

Boolean HasLocalChanges
Returns True if the linked drawing contains data for one or more editing conflicts. Read only.

Number ID
Returns unique component ID. Read only.

Boolean IsCheckedOut
Checks if component is shared and checked out in Enterprise Edition. Read only.

Boolean IsCheckedOutRemotely
Checks if component is shared and checked out by another user in Enterprise Edition. Read only.

Boolean IsModified
Checks if component is modified since last save in Enterprise Edition. Read only.

Boolean IsOutdated
Checks if component is shared and there is a later version of the component on the Enterprise server in Enterprise Edition. Read only.

Boolean IsReadOnly
Checks if component is readonly (eg, shared and not checked out) in Enterprise Edition. Read only.
Boolean IsShared
Checks if component is shared on an Enterprise server in Enterprise Edition. Read only.

Boolean IsWritable
Checks if component is writable in Enterprise Edition. Read only.

DateTime LastRefreshed
Returns date and time of the last refresh for a linked component and the current date and time for a non-linked component.

Format LineBackground
Returns background color used for lines. Read only.

Format LineForeground
Returns foreground color used for lines. Read only.

Format LineSize
Returns size used for lines. Read only.

FormatLineStyle
Returns style used for lines. Read only.

Rect LinkArea
Returns the area of interest used by the linked drawing. If the drawing is not linked or does not use the area of interest, returns a rect with negative width or height. Read only.

Boolean LinkAreaRestrictive
Returns True if the linked drawing uses the area of interest and requires all objects to be completely within that area. Read only.

LinkEditMode LinkEditMode
Sets or returns the editing mode for the linked drawing. Read / write.

String LinkRowset
Returns the source rowset of a linked component. Read only.

String LinkSource
Returns the data source of a linked component. Read only.

String LinkTechnology
Returns the data access technology used by a linked component. Read only.

String Name
Sets or returns component name. Read/write.

String Note
Sets or returns notes associated with the component. Read/write.

ObjectSet ObjectSet
Returns set of drawing objects. Read only.

ComponentSet OwnedComponentSet
Returns set of components bound to this component. Read only.
ComponentSet OwnedLabelSet
Returns set of label components bound to this component. Read only.

ComponentSet OwnedLayoutSet
Returns set of layouts bound to this component. Read only.

Table OwnedTable
Returns table bound to this component. Read only.

Component Owner
Returns owner component if any. Read only.

Any Parent
Returns parent object. Read only.

Format PointBackground
Returns background color used for points. Read only.

Format PointForeground
Returns foreground color used for points. Read only.

Format PointRotation
Returns rotation used for points. Read only.

Format PointSize
Returns size used for points. Read only.

Format PointStyle
Returns style used for points. Read only.

Number RemoteVersion
Returns version of shared component available on the Enterprise server in Enterprise Edition. Read only.

DrawingSelectionSet SavedSelectionSet
Returns set of saved selections. Read only.

ObjectSet Selection
Returns set of selected drawing objects. Read only.

String Server
Returns server identification string for shared component in Enterprise Edition. Read only.

String ShareStatus
Returns textual report on component sharing status in Enterprise Edition. Read only.

ComponentType Type
Returns component type. Read only.

String TypeName
Returns textual representation of the component type. Read only.

Number Version
Returns version of shared component within the local project in Enterprise Edition. Read only.

ViewSet ViewSet
Returns set of views associated with this component. Read only.

Number ZoomMax
Sets or returns maximum zoom level. Read/write.

Number ZoomMin
Sets or returns minimum zoom level. Read/write.

Number ZoomRender
Sets or returns render zoom level. Read/write.

Methods

CheckIn()
Checks in changes made to a shared component to the Enterprise server in Enterprise Edition.

CheckOut()
Checks out a shared component for editing in Enterprise Edition.

Clear(Boolean selection)
Clears entire component or selection within the component. Calling the method without parameters clears entire component.

Boolean CommitLocalChanges()
Commits local changes to objects participating in editing conflicts in the linked drawing and uploads them to the data source. Returns True if all editing conflicts are resolved and False otherwise.

Copy(Boolean selection)
Copies entire component or selection within the component into the Clipboard. Calling the method without parameters copies entire component.

Cut(Boolean selection)
Copies entire component or selection within the component into the Clipboard and then clears it. Calling the method without parameters cuts entire component.

Boolean DiscardLocalChanges()
Discards local changes to objects participating in editing conflicts in the linked drawing and retrieves the latest data for these objects from the data source. Returns True if all editing conflicts are resolved and False otherwise.

GetLatestVersion()
Gets the latest version of a shared component from the Enterprise server in Enterprise Edition.

Boolean IsEmpty()
Returns True if drawing contains no objects and False otherwise.

Boolean IsLinked()
Returns True if drawing is linked and False otherwise.

Open()
Opens component.
OpenInNewWindow()
Opens component in new window.

Paste(Boolean replace)
Pastes the contents of the Clipboard into the component replacing current selection or leaving it intact. Calling the method without parameters replaces selection.

Print(Boolean useDialog)
Prints component.

ProjectTo(CoordinateSystem system)
Projects component to another coordinate system.

Boolean RefreshWithArea(Rect area, Boolean restrictive)
Refreshes a linked drawing using the specified area of interest. Does nothing if the drawing is not linked. Returns True if the refresh has succeeded and False otherwise.

RenderAreaTo(String name, Number width, Number height, Rect area, Boolean clean)
Renders component area into an image with given name, width and height. If last parameter is True, legend, north arrow and scale bar are not rendered even if they are on.

RenderTo(String name, Number width, Number height, Boolean clean)
Render component into an image with given name, width and height. If last parameter is True, legend, north arrow and scale bar are not rendered even if they are turned on.

Boolean Refresh()
Refreshes a linked component and does nothing for a non-linked component. In the case of a linked component returns True if the refresh has succeeded and False otherwise. Refreshing a table bound to a linked drawing refreshes the drawing and vice versa.

Boolean RefreshAfter(Number)
Refreshes a linked component if the component has not been refreshed for more than the given number of seconds.

SelectAll()
Selects all objects.

SelectInverse()
Selects objects that were not selected and vice versa.

SelectNone()
Unselects all objects.

UndoCheckOut()
Undoes changes made to a shared component and gets the latest version of the component from the Enterprise server in Enterprise Edition.

Unlink()
Unlinks component. If the component is not linked, does nothing.

Obtained From
Document (NewDrawing)

See Also
**DrawingSelectionSet Object**
Set of saved selections within drawing.

**Properties**

- **Application Application**
  Returns application. Read only.

- **Number Count**
  Returns number of selections in set. Read only.

- **ObjectSet Item(Number index)**
  Returns selection with given index. Read only.

- **Any Parent**
  Returns parent object. Read only.

**Methods**

- **Number ItemByMask(Number mask)**
  Returns index of selection with given mask value or -1.

- **Number ItemByName(String name)**
  Returns index of selection with given name or -1.

**Obtained From**

- **Drawing (SavedSelectionSet)**

**See Also**

- Scripting Reference
**DrawingWindow Object**

Drawing window.

**Properties**

Component `ActiveComponent`
Returns drawing component shown in window. Read only.

Object `ActiveObject`
Returns object selected for editing. Read only.

Application `Application`
Returns application. Read only.

Rect `Bounds`
Returns the area covered by the window, in the coordinate system of the displayed component. Read only.

Component `Component`
Returns drawing component shown in window. Read only.

Number `Height`
Returns window height in pixels. Read only.

Point `Location`
Sets or returns viewport center. Read/write.

Point `LocationLatLon`
Sets or returns viewport center in lat/lon coordinates. Read/write.

Any `Parent`
Returns parent object. Read only.

Number `Scale`
Sets or returns viewport scale (projected components only). Read/write.

Number `ScaleInternal`
Sets or returns internal pixel-to-data ratio. Read/write.

Number `Width`
Returns window width in pixels. Read only.

**Methods**

Close()
Closes window.

Boolean `HasActiveObject()`
Returns True if some object within the window is selected for editing and False otherwise.

Boolean `HasLocation()`
Returns True if viewport center can be modified and False otherwise.
Boolean HasScale()
Returns True if viewport scale can be obtained or modified and False otherwise.

Boolean HasScaleInternal()
Returns True.

MoveTo(Branch branch)
MoveTo(BranchSet branchSet)
MoveTo(Component component)
MoveTo(Geom geom)
MoveTo(Label label)
MoveTo(LabelSet labelSet)
MoveTo(Object object)
MoveTo(ObjectSet objectSet)
MoveTo(Pixel pixel)
MoveTo(PixelSet pixelSet)
MoveTo(Point point)
MoveTo(PointSet pointSet)
MoveTo(Rect rect)
Centers viewport on given object or set of objects. Branches, geoms, points, rects and their sets are assumed to be in the coordinate system of the displayed component.

MoveToLocation(Point point, Boolean latLon)
Centers viewport on given location. LatLon parameter is optional.

Refresh()
Updates window contents.

RenderTo(String name)
Renders window into an image with given name.

ZoomTo(Branch branch)
ZoomTo(BranchSet branchSet)
ZoomTo(Component component)
ZoomTo(Geom geom)
ZoomTo(Label label)
ZoomTo(LabelSet labelSet)
ZoomTo(Object object)
ZoomTo(ObjectSet objectSet)
ZoomTo(Pixel pixel)
ZoomTo(PixelSet pixelSet)
ZoomTo(Point point)
ZoomTo(PointSet pointSet)
ZoomTo(Rect rect)
Zooms to given object or set of objects. Branches, geoms, points, rects and their sets are assumed to be in the coordinate system of the displayed component. If the bounding box of the object is a point (for example, if the object is a point), centers viewport on given object.

Obtained From
WindowSet (Item, ActiveWindow)

See Also
**DSSAtom Object**
Atomic ranking expression used to build rank columns.

**Properties**

- **Application Application**
  Returns application. Read only.

- **Number Average**
  Returns average value used in atom. Read/write.

- **Column Column**
  Sets or returns owner column. Read/write.

- **Number Dispersion**
  Returns dispersion value used in atom. Read/write.

- **Number ID**
  Returns unique atom ID. Read only.

- **String Name**
  Sets or returns atom name. Read/write.

- **Any Parent**
  Returns parent object. Read only.

- **PointSet Points**
  Returns metric of the atom curve. Read only.

- **Boolean RefreshAverage**
  Sets or returns refresh option for average value. Read/write.

- **Boolean RefreshDispersion**
  Sets or returns refresh option for dispersion value. Read/write.

- **DSSAtomType Type**
  Returns atom type. Read only.

**Methods**

- **MakeAverage(Number average, Number dispersion)**
  Initializes atom with bell curve with given average and dispersion.

- **MakeCurve(PointSet points)**
  Initializes atom with arbitrary curve.

- **MakeHigh(Number average, Number dispersion)**
  Initializes atom with right elbow with given average and dispersion.

- **MakeLow(Number average, Number dispersion)**
  Initializes atom with left elbow with given average and dispersion.
Obtained From

DSSAtomSet (Item, LastAdded, NewDSSAtom)
DSSQueryEntry (Atom)

See Also

Scripting Reference
DSSAtomSet Object
Set of atomic ranking expressions used to build rank columns.

Properties

Application Application
Returns application. Read only.

Number Count
Returns number of atoms in set. Read only.

DSSAtom Item(Number index)
DSSAtom Item(String name)
Returns atom with given index or name. Read only.

DSSAtom LastAdded
Returns last added atom. Read only.

Any Parent
Returns parent object. Read only.

Methods

Add(DSSAtom atom)
Appends atom to atom set.

Number ItemByID(Number id)
Returns index of atom with given ID or -1.

Number ItemByName(String name)
Returns index of atom with given name or -1.

DSSAtom NewDSSAtom()
Returns new atom that can later be added to atom set.

Remove(Number index)
Removes atom with given index.

RemoveAll()
Removes all atoms.

Obtained From

Table (DSSAtomSet)

See Also

Scripting Reference
**DSSQuery Object**
Set of ranking expressions.

**Properties**

- Application Application
  Returns application. Read only.

- Number Count
  Returns number of expressions in set. Read only.

- DSSQueryEntry Item(Number index)
  Returns expression with given index. Read only.

- DSSQueryEntry LastAdded
  Returns last added expression if any. Read only.

- Any Parent
  Returns parent object. Read only.

**Methods**

- Add(DSSQueryEntry queryEntry)
  Appends ranking expression to set.

- DSSQueryEntry NewDSSQueryEntry()
  Returns new expression that can later be added to set.

- Remove(Number index)
  Removes expression with given index.

**Obtained From**

- Column (DSSQuery)

**See Also**

Scripting Reference
DSSQueryEntry Object
Ranking expression used to build rank columns.

Properties

Application Application
Returns application. Read only.

DSSAtom Atom
Sets or returns atom. Read/write.

DSSHedge Hedge
Sets or returns atom hedge. Read/write.

DSSJunction Junction
Sets or returns junction to next atom in query. Read/write.

Boolean Not
Sets or returns negation flag. Read/write.

Any Parent
Returns parent object. Read only.

Obtained From

DSSQuery (Item, LastAdded, NewDSSQueryEntry)

See Also

Scripting Reference
Elevation Object
Elevation component.

Properties

Application
Returns application. Read only.

Color
BackgroundColor
Sets or returns background color. Read / write.

Boolean
Cached
Sets or returns caching option for shared component in Enterprise Edition. Read/write.

Boolean
Coordinates
Turns coordinate display on or off. Read/write.

String
Description
Sets or returns component description. Read/write.

Boolean
DistanceLabels
Turns labels for coordinate distances on or off. Read/write.

Boolean
DistanceLabelsEven
Turns labels for even distances on or off. Read/write.

Number
DistanceLabelsDigits
Sets or returns the number of decimal digits used for distance labels. Read/write.

Document
Document
Returns owner document. Read only.

Folder
Folder
Sets or returns containing folder. Read/write.

Boolean
HeightLabels
Turns labels for coordinate heights on or off. Read/write.

Boolean
HeightLabelsBounds
Turns labels for minimum and maximum heights on or off. Read/write.

Boolean
HeightLabelsBreaks
Turns labels for surface breaks on or off. Read/write.

Boolean
HeightLabelsEven
Turns labels for even heights on or off. Read/write.

Number
HeightLabelsDigits
Sets or returns the number of decimal digits used for height labels. Read/write.

Number
ID
Returns unique component ID. Read only.
Boolean IsCheckedOut
Checks if component is shared and checked out in Enterprise Edition. Read only.

Boolean IsCheckedOutRemotely
Checks if component is shared and checked out by another user in Enterprise Edition. Read only.

Boolean IsModified
Checks if component is modified since last save in Enterprise Edition. Read only.

Boolean IsOutdated
Checks if component is shared and there is a later version of the component on the Enterprise server in Enterprise Edition. Read only.

Boolean IsReadOnly
Checks if component is readonly (eg, shared and not checked out) in Enterprise Edition. Read only.

Boolean IsShared
Checks if component is shared on an Enterprise server in Enterprise Edition. Read only.

Boolean IsWritable
Checks if component is writable in Enterprise Edition. Read only.

String Name
Sets or returns component name. Read/write.

String Note
Sets or returns notes associated with the component. Read/write.

ComponentSet OwnedComponentSet
Returns set of components bound to this component. Read only.

ComponentSet OwnedLayoutSet
Returns set of layouts bound to this component. Read only.

Component Owner
Returns owner component if any. Read only.

Any Parent
Returns parent object. Read only.

Number RemoteVersion
Returns version of shared component available on the Enterprise server in Enterprise Edition. Read only.

String Server
Returns server identification string for shared component in Enterprise Edition. Read only.

Boolean Shading
Turns color shading on or off. Read/write.

String ShareStatus
Returns textual report on component sharing status in Enterprise Edition. Read only.
ComponentType Type
Returns component type. Read only.

String TypeName
Returns textual representation of the component type. Read only.

Number Version
Returns version of shared component within the local project in Enterprise Edition. Read only.

Boolean ZeroHeight
Includes or excludes zero height. Read/write.

**Methods**

CheckIn()
Checks in changes made to a shared component to the Enterprise server in Enterprise Edition.

CheckOut()
Checks out a shared component for editing in Enterprise Edition.

Copy()
Copies component into the Clipboard.

GetLatestVersion()
Gets the latest version of a shared component from the Enterprise server in Enterprise Edition.

Open()
Opens component.

OpenInNewWindow()
Opens component in new window.

Print(Boolean useDialog)
Prints component.

RenderTo(String name, Number width, Number height)
Render component into an image with given name, width and height.

UndoCheckOut()
Undoes changes made to a shared component and gets the latest version of the component from the Enterprise server in Enterprise Edition.

**Obtained From**

Document (NewElevation)

**See Also**

Scripting Reference
Ellipsoid Object
Ellipsoid.

Properties

Application
Returns application. Read only.

Number Eccentricity
Returns eccentricity. Read/write.

Number MajorAxis
Returns major axis in meters. Read/write.

Number MinorAxis
Returns minor axis in meters. Read/write.

String Name
Returns ellipsoid name. Read only.

Any Parent
Returns parent object. Read only.

Methods

Load(String name)
Loads ellipsoid using given name.

Obtained From

Application (DefaultEllipsoid, NewEllipsoid)
Datum (Ellipsoid)
EllipsoidSet (Item)

See Also

Scripting Reference
EllipsoidSet Object
Set of ellipsoids.

Properties

Application
Application Application
Returns application. Read only.

Number Count
Returns number of ellipsoids in set. Read only.

Ellipsoid Item(Number index)
Ellipsoid Item(String name)
Returns ellipsoid with given index or name. Read only.

Any Parent
Returns parent object. Read only.

Methods

Number ItemByName(String name)
Returns index of ellipsoid with given name or -1.

Obtained From

Application (EllipsoidSet)

See Also

Scripting Reference
EventArgs Object
Arguments for application-wide events.

Properties

Application Application
Returns application object. Read only.

Boolean Handled
Sets or returns handled flag. Setting this property to True will cancel further processing of an event by the system. Read/write.

Label Label
Returns context label if any. Read only.

Point LocationLatLon
Returns event location in lat/lon coordinates. Read only.

Point LocationNative
Returns event location in coordinates of the context component. Read only.

Point LocationScreen
Returns event location in window coordinates. Read only.

Object Object
Returns context object if any. Read only.

Any Parent
Returns parent object. Read only.

Methods

Boolean HasLabel()
Returns True if the event has a context label and False otherwise.

Boolean HasObject()
Returns True if the event has a context object and False otherwise.

See Also

Scripting Reference
Export Object
Generic exporter.

Properties

Application Application
Returns application. Read only.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string (eg. "Text Files (*.txt)"). Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of export properties. Read only.

Methods

Boolean CanExport(ComponentType type)
Returns True if exporter can save components of given type and False otherwise.

Export(Component component, String fileName, ConvertPrompt prompt)
Exports given component to a file.

Obtained From

Application (NewExport)

See Also

Scripting Reference
ExportAdoNet Object

ADO.NET export.

Properties

Application Application
Returns application. Read only.

String ConnectionString
Sets or returns connection string. Read/write.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of export properties. Read only.

Methods

Boolean CanExport(ComponentType type)
Returns True if exporter can save components of given type and False otherwise.

Export(Component component, String fileName, ConvertPrompt prompt)
Exports given component to a file.

Obtained From

Application (NewExport("ADO.NET"))

See Also

Scripting Reference
ExportAdoNetOdbc Object
ADO.NET ODBC export.

**Properties**

Application Application
Returns application. Read only.

String ConnectionString
Sets or returns connection string. Read/write.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of export properties. Read only.

**Methods**

Boolean CanExport(ComponentType type)
Returns True if exporter can save components of given type and False otherwise.

Export(Component component, String fileName, ConvertPrompt prompt)
Exports given component to a file.

**Obtained From**

Application (NewExport("ADO.NET ODBC"))

**See Also**

Scripting Reference
ExportAdoNetOleDb Object

ADO.NET OLE DB export.

Properties

Application Application
  Returns application. Read only.

String ConnectionString
  Sets or returns connection string. Read/write.

String DefaultExtension
  Returns default file extension. Read only.

String Filter
  Returns filter string. Read only.

String Name
  Returns file format name. Read only.

Any Parent
  Returns parent object. Read only.

ConverterPropertySet PropertySet
  Returns set of export properties. Read only.

Methods

Boolean CanExport(ComponentType type)
  Returns True if exporter can save components of given type and False otherwise.

Export(Component component, String fileName, ConvertPrompt prompt)
  Exports given component to a file.

Obtained From

Application (NewExport("ADO.NET OLE DB"))

See Also

Scripting Reference
ExportAdoNetOracle Object

ADO.NET Oracle export.

Properties

Application Application
  Returns application. Read only.

String ConnectionString
  Sets or returns connection string. Read/write.

String DefaultExtension
  Returns default file extension. Read only.

String Filter
  Returns filter string. Read only.

String Name
  Returns file format name. Read only.

Any Parent
  Returns parent object. Read only.

ConverterPropertySet PropertySet
  Returns set of export properties. Read only.

Methods

String BuildConnectionString(String server, String login, String password)
  Composes a connection string for Oracle ADO.NET provider. The last two arguments are optional.

Boolean CanExport(ComponentType type)
  Returns True if exporter can save components of given type and False otherwise.

Export(Component component, String fileName, ConvertPrompt prompt)
  Exports given component to a file.

Obtained From

  Application (NewExport("ADO.NET Oracle"))

See Also

  Scripting Reference
**ExportAdoNetSqlServer Object**

ADO.NET SQL Server export.

**Properties**

Application Application
   Returns application. Read only.

String ConnectionString
   Sets or returns connection string. Read/write.

String DefaultExtension
   Returns default file extension. Read only.

String Filter
   Returns filter string. Read only.

String Name
   Returns file format name. Read only.

Any Parent
   Returns parent object. Read only.

ConverterPropertySet PropertySet
   Returns set of export properties. Read only.

**Methods**

String BuildConnectionString(String server, String login, String password, String database)
   Composes a connection string for SQL Server ADO.NET provider. The last three arguments are optional.

Boolean CanExport(ComponentType type)
   Returns True if exporter can save components of given type and False otherwise.

Export(Component component, String fileName, ConvertPrompt prompt)
   Exports given component to a file.

**Obtained From**

Application (NewExport("ADO.NET SQL Server"))

**See Also**

Scripting Reference
ExportAi Object
Adobe Illustrator AI exporter.

Properties

Application Application
Returns application. Read only.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of export properties. Read only.

Boolean VectorText
Toggles the conversion of text element into vector shapes. Read/write.

Methods

Boolean CanExport(ComponentType type)
Returns True if exporter can save components of given type and False otherwise.

Export(Component component, String fileName, ConvertPrompt prompt)
Exports given component to a file.

Obtained From

Application (NewExport("AI"))

See Also

Scripting Reference
ExportBil Object
BIL exporter.

Properties

Application Application
Returns application. Read only.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of export properties. Read only.

Methods

Boolean CanExport(ComponentType type)
Returns True if exporter can save components of given type and False otherwise.

Export(Component component, String fileName, ConvertPrompt prompt)
Exports given component to a file.

Obtained From

Application (NewExport("BIL"))

See Also

Scripting Reference
**ExportBmp Object**
BMP exporter.

**Properties**

Application Application
Returns application. Read only.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of export properties. Read only.

**Methods**

Boolean CanExport(ComponentType type)
Returns True if exporter can save components of given type and False otherwise.

Export(Component component, String fileName, ConvertPrompt prompt)
Exports given component to a file.

**Obtained From**

Application (NewExport("BMP"))

**See Also**

Scripting Reference
ExportCsv Object
CSV exporter.

**Properties**

Application Application
Returns application. Read only.

ConverterItemSet Columns
Returns set of export columns. Read only.

ConvertPolicy ConvertPolicy
Sets or returns column export policy. Read/write.

String DefaultExtension
Returns default file extension. Read only.

Boolean EndingComma
Sets or returns ending comma option. If option is set to True, the system writes an extra comma at the end of each line. Read/write.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of export properties. Read only.

Boolean Quotes
Sets or returns quoting option. If option is set to True, the system encloses values in quotation marks regardless of their type. Read/write.

**Methods**

Boolean CanExport(ComponentType type)
Returns True if exporter can save components of given type and False otherwise.

Export(Component component, String fileName, ConvertPrompt prompt)
Exports given component to a file.

**Obtained From**

Application (NewExport("CSV"))

**See Also**

Scripting Reference
**ExportDb Object**

DB export.

**Properties**

Application Application
Returns application. Read only.

ConverterItemSet Columns
Returns set of export columns. Read only.

ConvertPolicy ConvertPolicy
Sets or returns column export policy. Read/write.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of export properties. Read only.

SubtypeDb Subtype
Sets or returns subtype of export file. Read/write.

**Methods**

Boolean CanExport(ComponentType type)
Returns True if exporter can save components of given type and False otherwise.

Export(Component component, String fileName, ConvertPrompt prompt)
Exports given component to a file.

**Obtained From**

Application (NewExport("DB"))

**See Also**

Scripting Reference
ExportDb2 Object
Native DB2 export.

Properties

Application Application
  Returns application. Read only.

String ConnectionString
  Sets or returns connection string. Read/write.

String DefaultExtension
  Returns default file extension. Read only.

String Filter
  Returns filter string. Read only.

String Name
  Returns file format name. Read only.

Any Parent
  Returns parent object. Read only.

ConverterPropertySet PropertySet
  Returns set of export properties. Read only.

Methods

String BuildConnectionString(String server, String login, String password)
  Composes a connection string for DB2 native client. The last two arguments are optional.

Boolean CanExport(ComponentType type)
  Returns True if exporter can save components of given type and False otherwise.

Export(Component component, String connectionString, ConvertPrompt prompt)
  Exports given component.

Obtained From

Application (NewExport("DB2"))

See Also

Scripting Reference
ExportDbf Object

DBF export.

Properties

Application Application
Returns application. Read only.

ConverterItemSet Columns
Returns set of export columns. Read only.

ConvertPolicy ConvertPolicy
Sets or returns column export policy. Read/write.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of export properties. Read only.

SubtypeDBF Subtype
Sets or returns subtype of export file. Read/write.

Methods

Boolean CanExport(ComponentType type)
Returns True if exporter can save components of given type and False otherwise.

Export(Component component, String fileName, ConvertPrompt prompt)
Exports given component to a file.

Obtained From

Application (NewExport("DBF"))

See Also

Scripting Reference
ExportDxf Object
DXF export.

Properties

Application Application
Returns application. Read only.

ConverterItemSet Columns
Returns set of export columns. Read only.

String ColumnZ
Sets or returns the name of a column with height data. Read / write.

ConvertPolicy ConvertPolicy
Sets or returns column export policy. Read/write.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of export properties. Read only.

Methods

Boolean CanExport(ComponentType type)
Returns True if exporter can save components of given type and False otherwise.

Export(Component component, String fileName, ConvertPrompt prompt)
Exports given component to a file.

Obtained From

Application (NewExport("DXF"))

See Also

Scripting Reference
ExportE00 Object
E00 format exporter.

Properties

Application Application
Returns application. Read only.

ConverterItemSet Columns
Returns set of export columns. Read only.

ConvertPolicy ConvertPolicy
Sets or returns column export policy. Read/write.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of export properties. Read only.

Methods

Boolean CanExport(ComponentType type)
Returns True if exporter can save components of given type and False otherwise.

Export(Component component, String fileName, ConvertPrompt prompt)
Exports given component to a file.

Obtained From

Application (NewExport("E00"))

See Also

Scripting Reference
**ExportEcw Object**
ECW exporter.

**Properties**

- **Application Application**
  Returns application. Read only.

- **Number CompressionRatio**
  Sets or returns compression ratio. Read/write.

- **String DefaultExtension**
  Returns default file extension. Read only.

- **String Filter**
  Returns filter string. Read only.

- **String Name**
  Returns file format name. Read only.

- **Any Parent**
  Returns parent object. Read only.

- **ConverterPropertySet PropertySet**
  Returns set of export properties. Read only.

**Methods**

- **Boolean CanExport(ComponentType type)**
  Returns True if exporter can save components of given type and False otherwise.

- **Export(Component component, String fileName, ConvertPrompt prompt)**
  Exports given component to a file.

**Obtained From**

- Application (NewExport("ECW"))

**See Also**

Scripting Reference
ExportEmf Object
EMF exporter.

Properties

Application Application
Returns application. Read only.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of export properties. Read only.

Methods

Boolean CanExport(ComponentType type)
Returns True if exporter can save components of given type and False otherwise.

Export(Component component, String fileName, ConvertPrompt prompt)
Exports given component to a file.

Obtained From

Application (NewExport("EMF"))

See Also

Scripting Reference
ExportEmfGdi Object
EMF (GDI) exporter for layouts.

Properties

Application Application
Returns application object. Read only.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

Boolean IgnoreStyles
Toggles the use of style information for drawing objects and labels. Read/write.

String Name
Returns file format name. Read only.

Number Page
Sets or returns exported page for a multipage layout. Read/write.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of export properties. Read only.

Number Resolution
Sets or returns pixel resolution in dots per inch (DPI). Read/write.

Number VectorResolution
Sets or returns vector resolution in dots per inch (DPI). Read/write.

Methods

Boolean CanExport(ComponentType type)
Returns True if exporter can save components of given type and False otherwise.

Export(Component component, String fileName, ConvertPrompt prompt)
Exports given component to a file.

Obtained From

Application (NewExport("EMF (GDI)"))

See Also

Scripting Reference
**ExportEmfGdiPlus Object**
EMF (GDI+) exporter for layouts.

**Properties**

- **Application Application**
  Returns application object. Read only.

- **String DefaultExtension**
  Returns default file extension. Read only.

- **String Filter**
  Returns filter string. Read only.

- **Boolean IgnoreStyles**
  Toggles the use of style information for drawing objects and labels. Read/write.

- **String Name**
  Returns file format name. Read only.

- **Number Page**
  Sets or returns exported page for a multipage layout. Read/write.

- **Any Parent**
  Returns parent object. Read only.

- **ConverterPropertySet PropertySet**
  Returns set of export properties. Read only.

- **Number Resolution**
  Sets or returns pixel resolution in dots per inch (DPI). Read/write.

- **Number VectorResolution**
  Sets or returns vector resolution in dots per inch (DPI). Read/write.

**Methods**

- **Boolean CanExport(ComponentType type)**
  Returns True if exporter can save components of given type and False otherwise.

- **Export(Component component, String fileName, ConvertPrompt prompt)**
  Exports given component to a file.

**Obtained From**

- Application (NewExport("EMF (GDI+)"))

**See Also**

- Scripting Reference
ExportFit Object
ESRI float grid exporter.

Properties

Application Application
Returns application. Read only.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of export properties. Read only.

Methods

Boolean CanExport(ComponentType type)
Returns True if exporter can save components of given type and False otherwise.

Export(Component component, String fileName, ConvertPrompt prompt)
Exports given component to a file.

Obtained From

Application (NewExport("FLT") or NewExport("ESRI Float Grid"))

See Also

Scripting Reference
ExportGif Object
GIF exporter.

Properties

Application Application
Returns application. Read only.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of export properties. Read only.

Methods

Boolean CanExport(ComponentType type)
Returns True if exporter can save components of given type and False otherwise.

Export(Component component, String fileName, ConvertPrompt prompt)
Exports given component to a file.

Obtained From

Application (NewExport("GIF"))

See Also

Scripting Reference
ExportGrd Object
ESRI ASCII grid exporter.

Properties

Application Application
Returns application. Read only.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of export properties. Read only.

Methods

Boolean CanExport(ComponentType type)
Returns True if exporter can save components of given type and False otherwise.

Export(Component component, String fileName, ConvertPrompt prompt)
Exports given component to a file.

Obtained From

Application (NewExport("GRD") or NewExport("ESRI ASCII Grid"))

See Also

Scripting Reference
ExportGrdSurfer6 Object
Surfer 6 GRD exporter.

Properties

Application Application
Returns application. Read only.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of export properties. Read only.

Methods

Boolean CanExport(ComponentType type)
Returns True if exporter can save components of given type and False otherwise.

Export(Component component, String fileName, ConvertPrompt prompt)
Exports given component to a file.

Obtained From

Application (NewExport("Surfer 6 GRD"))

See Also

Scripting Reference
**ExportGrdSurfer7 Object**

Surfer 7 GRD exporter.

**Properties**

- **Application Application**
  Returns application. Read only.

- **String DefaultExtension**
  Returns default file extension. Read only.

- **String Filter**
  Returns filter string. Read only.

- **String Name**
  Returns file format name. Read only.

- **Any Parent**
  Returns parent object. Read only.

- **ConverterPropertySet PropertySet**
  Returns set of export properties. Read only.

**Methods**

- **Boolean CanExport(ComponentType type)**
  Returns True if exporter can save components of given type and False otherwise.

- **Export(Component component, String fileName, ConvertPrompt prompt)**
  Exports given component to a file.

**Obtained From**

- Application (NewExport("Surfer 7 GRD"))

**See Also**

- Scripting Reference
ExportGrdSurferAscii Object
Surfer ASCII GRD exporter.

Properties

Application Application
Returns application. Read only.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of export properties. Read only.

Methods

Boolean CanExport(ComponentType type)
Returns True if exporter can save components of given type and False otherwise.

Export(Component component, String fileName, ConvertPrompt prompt)
Exports given component to a file.

Obtained From

Application (NewExport("Surfer ASCII GRD"))

See Also

Scripting Reference
ExportHtml Object
HTML exporter.

Properties

Application Application
Returns application. Read only.

ConverterItemSet Columns
Returns set of export columns. Read only.

ConvertPolicy ConvertPolicy
Sets or returns column export policy. Read/write.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of export properties. Read only.

Methods

Boolean CanExport(ComponentType type)
Returns True if exporter can save components of given type and False otherwise.

Export(Component component, String fileName, ConvertPrompt prompt)
Exports given component to a file.

Obtained From

Application (NewExport("HTML"))

See Also

Scripting Reference
ExportJpeg Object
JPEG exporter.

Properties

Application Application
Returns application. Read only.

JpegCompression Compression
Sets or returns JPEG compression method. Read/write.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of export properties. Read only.

JpegQuality Quality
Sets or returns JPEG compression quality. Read/write.

Number Scans
Sets or returns the number of scans used for compression. Read/write.

Methods

Boolean CanExport(ComponentType type)
Returns True if exporter can save components of given type and False otherwise.

Export(Component component, String fileName, ConvertPrompt prompt)
Exports given component to a file.

Obtained From

Application (NewExport("JPEG"))

See Also

Scripting Reference
ExportKml Object
KML export.

Properties

Application Application
Returns application. Read only.

String ColumnDescription
Sets or returns the name of a column with object descriptions. Read/write.

String ColumnName
Sets or returns the name of a column with object names. Read/write.

String ColumnZ
Sets or returns the name of a column with height data. Read/write.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

Boolean RelativeZ
Sets or returns the value of the relative heights option used with height data. If True, heights are relative, otherwise they are absolute. Read/write.

Methods

Boolean CanExport(ComponentType type)
Returns True if exporter can save components of given type and False otherwise.

Export(Component component, String filename, ConvertPrompt prompt)
Exports given component to a file.

Obtained From

Application (NewExport("KML "))

See Also

Scripting Reference
ExportMdb Object

MDB exporter.

Properties

Application
Returns application. Read only.

Columns
Returns set of export columns. Read only.

ConvertPolicy
Sets or returns column export policy. Read/write.

DefaultExtension
Returns default file extension. Read only.

Filter
Returns filter string. Read only.

Name
Returns file format name. Read only.

Parent
Returns parent object. Read only.

PropertySet
Returns set of export properties. Read only.

Subtype
Sets or returns subtype of export file. Read/write.

Methods

CanExport(ComponentType type)
Returns True if exporter can save components of given type and False otherwise.

Export(Component component, String fileName, ConvertPrompt prompt)
Exports given component to a file.

Obtained From

Application (NewExport("MDB"))

See Also

Scripting Reference
ExportMfd Object
MFD exporter.

Properties

Application Application
Returns application. Read only.

ConverterItemSet Columns
Returns set of export columns. Read only.

ConvertPolicy ConvertPolicy
Sets or returns column export policy. Read/write.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of export properties. Read only.

Methods

Boolean CanExport(ComponentType type)
Returns True if exporter can save components of given type and False otherwise.

Export(Component component, String fileName, ConvertPrompt prompt)
Exports given component to a file.

Obtained From

Application (NewExport("MFD"))

See Also

Scripting Reference
ExportMif Object
MIF exporter.

Properties

Application Application
Returns application. Read only.

ConverterItemSet Columns
Returns set of export columns. Read only.

ConvertPolicy ConvertPolicy
Sets or returns column export policy. Read/write.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of export properties. Read only.

Methods

Boolean CanExport(ComponentType type)
Returns True if exporter can save components of given type and False otherwise.

Export(Component component, String fileName, ConvertPrompt prompt)
Exports given component to a file.

Obtained From

Application (NewExport("MIF"))

See Also

Scripting Reference
ExportOdbc Object
ODBC export.

Properties

Application Application
  Returns application. Read only.

String ConnectionString
  Sets or returns connection string. Read/write.

String DefaultExtension
  Returns default file extension. Read only.

String Filter
  Returns filter string. Read only.

String Name
  Returns file format name. Read only.

Any Parent
  Returns parent object. Read only.

ConverterPropertySet PropertySet
  Returns set of export properties. Read only.

Methods

Boolean CanExport(ComponentType type)
  Returns True if exporter can save components of given type and False otherwise.

Export(Component component, String fileName, ConvertPrompt prompt)
  Exports given component to a file.

Obtained From

Application (NewExport("ODBC"))

See Also

Scripting Reference
ExportOleDb Object
OLE DB export.

Properties

Application Application
Returns application. Read only.

String ConnectionString
Sets or returns connection string. Read/write.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of export properties. Read only.

Methods

Boolean CanExport(ComponentType type)
Returns True if exporter can save components of given type and False otherwise.

Export(Component component, String fileName, ConvertPrompt prompt)
Exports given component to a file.

Obtained From

Application (NewExport("OLEDB"))

See Also

Scripting Reference
ExportOracle Object
Native Oracle export.

Properties

Application Application
   Returns application. Read only.

String ConnectionString
   Sets or returns connection string. Read/write.

String DefaultExtension
   Returns default file extension. Read only.

String Filter
   Returns filter string. Read only.

String Name
   Returns file format name. Read only.

Any Parent
   Returns parent object. Read only.

ConverterPropertySet PropertySet
   Returns set of export properties. Read only.

Methods

String BuildConnectionString(String server, String login, String password)
   Composes a connection string for Oracle native client. The last two arguments are optional.

Boolean CanExport(ComponentType type)
   Returns True if exporter can save components of given type and False otherwise.

Export(Component component, String connectionString, ConvertPrompt prompt)
   Exports given component.

Obtained From

Application (NewExport("Oracle"))

See Also

Scripting Reference
ExportPdf Object
PDF exporter for layouts.

Properties

Application Application
Returns application object. Read only.

Boolean Compression
Toggles compression. Read/write.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

Boolean IgnoreStyles
Toggles the use of style information for drawing objects and labels. Read/write.

Boolean Layers
Toggles the generation of layers. Read/write.

String Name
Returns file format name. Read only.

String PageFilter
Sets or returns page filter (eg, "1-5,8"). Read/write.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of export properties. Read only.

Number Resolution
Sets or returns pixel resolution in dots per inch (DPI). Read/write.

Boolean Transparency
Toggles support for transparency. Read/write.

Number VectorResolution
Sets or returns vector resolution in dots per inch (DPI). Read/write.

Boolean VectorText
Toggles the conversion of text element into vector shapes. Read/write.

Methods

Boolean CanExport(ComponentType type)
Returns True if exporter can save components of given type and False otherwise.
Export(Component component, String fileName, ConvertPrompt prompt)
Exports given component to a file.

**Obtained From**

Application (NewExport("PDF"))

**See Also**

Scripting Reference
ExportPng Object
PNG exporter.

Properties

Application
Returns application. Read only.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

Boolean Interlaced
Sets or returns interlaced option. Read/write.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet
Returns set of export properties. Read only.

Methods

Boolean CanExport(ComponentType type)
Returns True if exporter can save components of given type and False otherwise.

Export(Component component, String fileName, ConvertPrompt prompt)
Exports given component to a file.

Obtained From

Application (NewExport("PNG"))

See Also

Scripting Reference
**ExportPostgreSQL Object**
Native PostgreSQL export.

**Properties**

- **Application Application**
  Returns application. Read only.

- **ConnectionString**
  Sets or returns connection string. Read/write.

- **DefaultExtension**
  Returns default file extension. Read only.

- **Filter**
  Returns filter string. Read only.

- **Name**
  Returns file format name. Read only.

- **Parent**
  Returns parent object. Read only.

- **PropertySet**
  Returns set of export properties. Read only.

**Methods**

- **BuildConnectionString(String server, String login, String password, String database)**
  Composes a connection string for PostgreSQL native client. All arguments except server are optional.

- **CanExport(ComponentType type)**
  Returns True if exporter can save components of given type and False otherwise.

- **Export(Component component, String connectionString, ConvertPrompt prompt)**
  Exports given component.

**Obtained From**

- Application (NewExport("PostgreSQL"))

**See Also**

- [Scripting Reference](#)
ExportPs Object

PS exporter for layouts.

Properties

Application Application
  Returns application object. Read only.

String DefaultExtension
  Returns default file extension. Read only.

Boolean ExplicitMask
  Toggles the use of explicit image mask for invisible pixels. Read/write.

String Filter
  Returns filter string. Read only.

Boolean IgnoreStyles
  Toggles the use of style information for drawing objects and labels. Read/write.

String Name
  Returns file format name. Read only.

String PageFilter
  Sets or returns page filter (eg, "1-5,8"). Read/write.

Any Parent
  Returns parent object. Read only.

PsPixelEncoding PixelEncoding
  Sets or returns pixel encoding. Read/write.

ConverterPropertySet PropertySet
  Returns set of export properties. Read only.

Number Resolution
  Sets or returns pixel resolution in dots per inch (DPI). Read/write.

Number VectorResolution
  Sets or returns vector resolution in dots per inch (DPI). Read/write.

Methods

Boolean CanExport(ComponentType type)
  Returns True if exporter can save components of given type and False otherwise.

Export(Component component, String fileName, ConvertPrompt prompt)
  Exports given component to a file.

Obtained From

Application (NewExport("PS"))
See Also

Scripting Reference
ExportRawBinary Object
Raw binary exporter.

Properties

Application Application
Returns application. Read only.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of export properties. Read only.

Methods

Boolean CanExport(ComponentType type)
Returns True if exporter can save components of given type and False otherwise.

Export(Component component, String fileName, ConvertPrompt prompt)
Exports given component to a file.

Obtained From

Application (NewExport("Raw Binary"))

See Also

Scripting Reference
ExportSdts Object
SDTS exporter.

Properties

Application Application
Returns application. Read only.

ConverterItemSet Columns
Returns set of export columns. Read only.

ConvertPolicy ConvertPolicy
Sets or returns column export policy. Read/write.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

Number LineLimit
Sets or returns maximum number of coordinates per line object in range from 100 to 5000. Default value is 2000. Read/write.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of export properties. Read only.

Methods

Boolean CanExport(ComponentType type)
Returns True if exporter can save components of given type and False otherwise.

Export(Component component, String fileName, ConvertPrompt prompt)
Exports given component to a file.

Obtained From

Application (NewExport("SDTS"))

See Also

Scripting Reference
**ExportShp Object**

SHP exporter.

**Properties**

Application Application  
Returns application. Read only.

ConverterItemSet Columns  
Returns set of export columns. Read only.

ConvertPolicy ConvertPolicy  
Sets or returns column export policy. Read/write.

String DefaultExtension  
Returns default file extension. Read only.

String Filter  
Returns filter string. Read only.

String Name  
Returns file format name. Read only.

Any Parent  
Returns parent object. Read only.

ConverterPropertySet PropertySet  
Returns set of export properties. Read only.

**Methods**

Boolean CanExport(ComponentType type)  
Returns True if exporter can save components of given type and False otherwise.

Export(Component component, String fileName, ConvertPrompt prompt)  
Exports given component to a file.

**Obtained From**

Application (NewExport("SHP"))

**See Also**

Scripting Reference
ExportSqlServer Object

SQL Server export.

Properties

Application Application
  Returns application. Read only.

String ConnectionString
  Sets or returns connection string. Read/write.

String DefaultExtension
  Returns default file extension. Read only.

String Filter
  Returns filter string. Read only.

String Name
  Returns file format name. Read only.

Any Parent
  Returns parent object. Read only.

ConverterPropertySet PropertySet
  Returns set of export properties. Read only.

Methods

String BuildConnectionString(String server, String login, String password, String database)
  Composes a connection string for SQL Server client. All arguments except server are optional.

Boolean CanExport(ComponentType type)
  Returns True if exporter can save components of given type and False otherwise.

Export(Component component, String connectionString, ConvertPrompt prompt)
  Exports given component.

Obtained From

Application (NewExport ("SQL Server"))

See Also

Scripting Reference
ExportTga Object
TGA exporter.

Properties

Application Application
Returns application. Read only.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of export properties. Read only.

Methods

Boolean CanExport(ComponentType type)
Returns True if exporter can save components of given type and False otherwise.

Export(Component component, String fileName, ConvertPrompt prompt)
Exports given component to a file.

Obtained From

Application (NewExport("TGA"))

See Also

Scripting Reference
ExportTiff Object
TIFF exporter.

Properties

Application Application
Returns application. Read only.

ByteOrder ByteOrder
Sets or returns byte order. Read/write.

TiffCompression Compression
Sets or returns compression method. Read/write.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of export properties. Read only.

Boolean UseProjection
Sets or returns projection export option. Read/write.

Methods

Boolean CanExport(ComponentType type)
Returns True if exporter can save components of given type and False otherwise.

Export(Component component, String fileName, ConvertPrompt prompt)
Exports given component to a file.

Obtained From

Application (NewExport("TIFF"))

See Also

Scripting Reference
ExportTxt Object
TXT exporter.

Properties

Application Application
Returns application. Read only.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of export properties. Read only.

Methods

Boolean CanExport(ComponentType type)
Returns True if exporter can save components of given type and False otherwise.

Export(Component component, String fileName, ConvertPrompt prompt)
Exports given component to a file.

Obtained From

Application (NewExport("TXT"))

See Also

Scripting Reference
ExportWk Object
WK exporter.

Properties

Application Application
Returns application.  Read only.

ConverterItemSet Columns
Returns set of export columns.  Read only.

ConvertPolicy ConvertPolicy
Sets or returns column export policy.  Read/write.

String DefaultExtension
Returns default file extension.  Read only.

String Filter
Returns filter string.  Read only.

String Name
Returns file format name.  Read only.

Any Parent
Returns parent object.  Read only.

ConverterPropertySet PropertySet
Returns set of export properties.  Read only.

SubtypeWk Subtype
Sets or returns subtype of export file.  Read/write.

Methods

Boolean CanExport(ComponentType type)
Returns True if exporter can save components of given type and False otherwise.

Export(Component component, String fileName, ConvertPrompt prompt)
Exports given component to a file.

Obtained From

Application (NewExport("WK"))

See Also

Scripting Reference
ExportXls Object
XLS exporter.

Properties

Application Application
Returns application. Read only.

ConverterItemSet Columns
Returns set of export columns. Read only.

ConvertPolicy ConvertPolicy
Sets or returns column export policy. Read/write.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of export properties. Read only.

SubtypeXls Subtype
Sets or returns subtype of export file. Read/write.

Methods

Boolean CanExport(ComponentType type)
Returns True if exporter can save components of given type and False otherwise.

Export(Component component, String fileName, ConvertPrompt prompt)
Exports given component to a file.

Obtained From

Application (NewExport("XLS"))

See Also

Scripting Reference
ExportXyz Object

XYZ exporter.

Properties

Application Application
  Returns application. Read only.

String DefaultExtension
  Returns default file extension. Read only.

String Filter
  Returns filter string. Read only.

String Name
  Returns file format name. Read only.

Any Parent
  Returns parent object. Read only.

ConverterPropertySet PropertySet
  Returns set of export properties. Read only.

Methods

Boolean CanExport(ComponentType type)
  Returns True if exporter can save components of given type and False otherwise.

Export(Component component, String fileName, ConvertPrompt prompt)
  Exports given component to a file.

Obtained From

Application (NewExport("XYZ"))

See Also

Scripting Reference
**Folder Object**
Folder component.

**Properties**

- **Application Application**
  Returns application. Read only.

- **ComponentSet Children**
  Returns set of contained components. Read only.

- **String Description**
  Sets or returns component description. Read/write.

- **Document Document**
  Returns owner document. Read only.

- **Folder Folder**
  Sets or returns containing folder. Read/write.

- **Number ID**
  Returns unique component ID. Read only.

- **String Name**
  Sets or returns component name. Read/write.

- **String Note**
  Sets or returns notes associated with the component. Read/write.

- **Component Owner**
  Returns owner component if any. Read only.

- **Any Parent**
  Returns parent object. Read only.

- **ComponentType Type**
  Returns component type. Read only.

- **String TypeName**
  Returns textual representation of the component type. Read only.

**Methods**

- **Boolean IsEmpty()**
  Returns True if folder does not contain any components and False otherwise.

- **Open()**
  Opens component.

- **OpenInNewWindow()**
  Opens component in new window.
Obtained From

Chart (Folder)
Comments (Folder)
Component (Folder)
Document (NewFolder)
Drawing (Folder)
Folder (Folder)
Form (Folder)
Image (Folder)
Labels (Folder)
Layout (Folder)
Map (Folder)
Palette (Folder)
Query (Folder)
Script (Folder)
Surface (Folder)
Table (Folder)
Terrain (Folder)
Zones (Folder)

See Also

Scripting Reference
Font Object

Font.

Properties

Boolean Bold
Sets or returns font bold option. Read/write.

Number Charset
Sets or returns font character set. Read/write.

Boolean Italic
Sets or returns font italic option. Read/write.

String Name
Sets or returns font facename. Read/write.

Number Size
Sets or returns font size in points. Read/write.

Boolean Strikethrough
Sets or returns font strikethrough option. Read/write.

Boolean Underline
Sets or returns font underline option. Read/write.

Number Weight
Sets or returns font weight. Read/write.

Methods

Font Clone()
Returns a copy of subject font.

Boolean IsEqual(Font font)
Returns True if subject font is equal to given font and False otherwise.

Events

FontChanged(String property)
Called when font has changed.

Obtained From

Font (Clone)
LayoutEntry (BorderFont, Font)

See Also

Scripting Reference
Form Object
Form component.

Properties

Application
Returns application. Read only.

String Description
Sets or returns component description. Read/write.

Document
Returns owner document. Read only.

Folder
Sets or returns containing folder. Read/write.

Number ID
Returns unique component ID. Read only.

String Name
Sets or returns component name. Read/write.

String Note
Sets or returns notes associated with the component. Read/write.

ComponentSet
Returns set of components bound to this component. Read only.

Component Owner
Returns owner component if any. Read only.

Any Parent
Returns parent object. Read only.

Any Result
Sets or returns form exit code. Read/write.

ComponentType
Returns component type. Read only.

String TypeName
Returns textual representation of the component type. Read only.

Methods

DoEvents()
Handles all pending events.

Open()
Opens component.
OpenInNewWindow()
Opens component in new window.

Run()
Runs form.

**Obtained From**

Document (NewForm)

**Comments**

Due to potential deadlock issues, it is recommended that one clears the Result property after invoking the form if it is known to contain a reference to an object, like a Component or Application object.

The DoEvents method is used to handle all pending events, for example, to allow canceling a lengthy operation.

**See Also**

Scripting Reference
Format Object
Set of formatting rules.

Properties

Application Application
Returns application. Read only.

Column Column
Returns column of a thematic formatting. Read only.

Component Component
Returns owner component. Read only.

FormatValue DefaultValue
Returns default format value. Read only.

FormatValue MaxValue
Returns format value applied to highest interval in interval-based thematic formatting. Read only.

FormatValue MinValue
Returns format value applied to lowest interval in interval-based thematic formatting. Read only.

Any Parent
Returns parent object. Read only.

FormatType Type
Returns formatting type. Read only.

FormatValueSet Values
Returns a set of format values. Read only.

Methods

Boolean IsConstant()
Returns True if formatting is not based on a column and False otherwise.

Boolean IsInitialized()
Returns True if formatting is initialized and False otherwise.

LoadFrom(String xml, Boolean loadColumn)
Loads formatting from given XML code. The last parameter is optional. If it is set to False or omitted, column data is not loaded.

LoadFromFile(String path, Boolean loadColumn)
Loads formatting from given XML file. The last parameter is optional. If it is set to False or omitted, column data is not loaded.

SaveToFile(String path)
Saves formatting to given XML file. If file already exists, it gets overwritten.

Set(Color value)
Set(Number value)
Unbinds formatting from any column it may have been bound to and sets the default format value to given color or number.

SetEqCount(Column column, Number breaks, Number alignment)
SetEqCount(Number id, Number breaks, Number alignment)
SetEqCount(String name, Number breaks, Number alignment)

Binds formatting to given column using "Equal Count" with specified number of intervals. Column can be supplied directly or by its ID or name. The last parameter is the decimal alignment (a value of 1 aligns numeric values to 10, a value of -1 aligns to 0.1, a value of -2 aligns to 0.01, the default value is 0 and aligns to whole numbers).

SetEqIntervals(Column column, Number breaks, Number alignment)
SetEqIntervals(Number id, Number breaks, Number alignment)
SetEqIntervals(String name, Number breaks, Number alignment)

Binds formatting to given column using "Equal Intervals" with specified number of intervals. Column can be supplied directly or by its ID or name. The last parameter is the decimal alignment (a value of 1 aligns numeric values to 10, a value of -1 aligns to 0.1, a value of -2 aligns to 0.01, the default value is 0 and aligns to whole numbers).

SetExpIntervals(Column column, Number breaks, Number alignment)
SetExpIntervals(Number id, Number breaks, Number alignment)
SetExpIntervals(String name, Number breaks, Number alignment)

Binds formatting to given column using "Exponential Intervals" with specified number of intervals. Column can be supplied directly or by its ID or name. The last parameter is the decimal alignment (a value of 1 aligns numeric values to 10, a value of -1 aligns to 0.1, a value of -2 aligns to 0.01, the default value is 0 and aligns to whole numbers).

SetNaturalBreaks(Column column, Number breaks, Number alignment)
SetNaturalBreaks(Number id, Number breaks, Number alignment)
SetNaturalBreaks(String name, Number breaks, Number alignment)

Binds formatting to given column using "Natural Breaks" with specified number of breaks. Column can be supplied directly or by its ID or name. The last parameter is the decimal alignment (a value of 1 aligns numeric values to 10, a value of -1 aligns to 0.1, a value of -2 aligns to 0.01, the default value is 0 and aligns to whole numbers).

SetStDevIntervals(Column column, Number breaks, Number alignment)
SetStDevIntervals(Number id, Number breaks, Number alignment)
SetStDevIntervals(String name, Number breaks, Number alignment)

Binds formatting to given column using "Standard Intervals" with specified number of intervals. Column can be supplied directly or by its ID or name. The last parameter is the decimal alignment (a value of 1 aligns numeric values to 10, a value of -1 aligns to 0.1, a value of -2 aligns to 0.01, the default value is 0 and aligns to whole numbers).

SetUniqueValues(Column column)
SetUniqueValues(Number id)
SetUniqueValues(String name)

Binds formatting to given column using "Unique Values". Column can be supplied directly or by its ID or name.

String ToXML()
Saves formatting to XML string.

Obtained From

Drawing (AreaBackground, AreaForeground, AreaSize, AreaStyle, LineBackground, LineForeground, LineSize, LineStyle, PointBackground, PointForeground, PointRotation, PointSize, PointStyle)
Labels (LabelBackground, LabelForeground, LabelRotation, LabelSize, LabelStyle)

See Also
Scripting Reference
FormatValue Object
Format value representing single color, size or style.

Properties

Application Application
Returns application. Read only.

Color Formatting or

Number Formatting
Sets or returns formatting (color or number). Read / write.

Any Parent
Returns parent object. Read only.

String Value
Returns unique value the formatting is bound to. Read only.

Number ValueLower
Returns numeric value representing the lower bound of the formatting interval. Read only.

Number ValueUpper
Returns numeric value representing the upper bound of the formatting interval. Read only.

Obtained From

FormatValueSet (Item property)
Format (DefaultValue, MinValue, MaxValue)

See Also

Scripting Reference
**FormatValueSet Object**
Set of format values for a thematic formatting.

**Properties**

Application Application
Returns application. Read only.

Number Count
Returns the number of format values. Read only.

FormatValue Item(Number index)
Returns format value with given index. Read only.

Any Parent
Returns parent object. Read only.

**Obtained From**

Format (Values)

**See Also**

Scripting Reference
**GeocodeMatch Object**
Geocoding match.

**Properties**

String Address
Returns match street address. Read only.

Application Application
Returns application. Read only.

String City
Returns match city. Read only.

String Country
Returns match country. Read only.

Number Latitude
Returns match latitude on a WGS84 datum. Read only.

GeocodeLevel Level
Returns match level. Read only.

String LevelText
Returns textual description of match level. Read only.

Number Longitude
Returns match longitude on a WGS84 datum. Read only.

Any Parent
Returns parent object. Read only.

String State
Returns match state. Read only.

GeocodeStatus Status
Returns match status. Read only.

String StatusText
Returns textual description of match status. Read only.

String Zip
Returns match zip code. Read only.

**Obtained From**

GeocodeMatchSet (Item)

**See Also**

Scripting Reference
**GeocodeMatchSet Object**
Set of geocoding matches.

**Properties**

Application Application
  Returns application. Read only.

Number Count
  Returns number of matches in set. Read only.

GeocodeMatch Item(Number index)
  Returns match with specified index. Read only.

Any Parent
  Returns parent object. Read only.

**Obtained From**

Geocoder (LocateAddress, LocateAddressRaw)

**See Also**

Scripting Reference
**Geocoder Object**

Geocoding engine.

**Properties**

Application Application

Returns application. Read only.

Any Parent

Returns parent object. Read only.

**Methods**

GeocodeMatchSet LocateAddress(String address, String city, String state, String zip, Number offset)

Geocodes an address given by address, city, state and zip strings. Multiple matches are ordered from the most likely to least likely. Offset parameter is specified in meters and can be omitted.

GeocodeMatchSet LocateAddressRaw(String address, Number offset)

Geocodes an address given by an unparsed address string. Multiple matches are ordered from the most likely to least likely. Offset parameter is specified in meters and can be omitted.

Point LocateZip(Number zip)

Point LocateZip(String zip)

Returns a lat/lon centroid for a given zip code.

**Obtained From**

Application (NewGeocoder)

**See Also**

Scripting Reference
**Geom Object**
Geometric entity.

**Properties**

**Application**
Returns application. Read only.

**Number Area**
Returns area measured over an ellipsoid. Returned value is in square meters. Read only.

**Number AreaNative**
Returns area in native units. Read only.

**Rect Box**
Sets or returns bounding box. Modifying this property scales and shifts the entity. Read/write.

**BranchSet**
Sets or returns set of branches (can accept BranchSet, Branch, PointSet or Point when setting). Read/write.

**Point Center**
Sets or returns center of minimum enclosing circle. Modifying this property moves the entity. Read/write.

**Point CenterInner**
Returns inner centroid for areas. Read only.

**Point CenterWeight**
Returns center of weight for areas. Read only.

**Number Length**
Returns length or perimeter measured over an ellipsoid. Returned value is in meters. Read only.

**Number LengthNative**
Returns length or perimeter in native units. Read only.

**Any Parent**
Returns parent object. Read only.

**GeomType**
Returns type of entity. Read only.

**Methods**

**Geom Buffer**
Returns a buffer of given width.

**Boolean CheckContains**
Returns True if entity contains another entity and False otherwise.

**Boolean CheckIntersects**
Returns True if entity intersects another entity and False otherwise.
Boolean CheckNeighbors(Geom geom, Number epsilon)
Returns True if entity neighbors another entity and False otherwise.

Boolean CheckTouches(Geom geom, Number epsilon)
Returns True if entity touches another entity and False otherwise.

GeomSet Decompose(Number epsilon)
Returns set of primitive parts composing geometric entity.

GeomSet DecomposeToConvexParts
Returns set of convex parts composing area.

GeomSet DecomposeToTriangles()
Returns set of triangles composing an area.

TriangleSet DecomposeToTrianglesAdv()
Returns set of triangles composing an area as a set of triplets identifying vertices for each triangle.

Number DistanceNative(Geom geom, Number epsilon)
Returns distance to given entity.

Number Distance(Geom geom, Number epsilon)
Returns distance to given entity in degrees or meters depending on whether or not the coordinate system of the entity is lat/lon or not.

Number DistanceToPoint(Point point)
Returns distance to given point.

Number EnclosingCircle(Point point)
Returns radius of minimum enclosing circle and puts center of this circle into supplied parameter.

Geom EnclosingRect()
Returns minimum enclosing rectangle.

Geom Intersect(GeomSet geomSet, Number epsilon)
Returns a copy of an entity intersected with given set of entities.

Geom Normalize (Number epsilon)
Returns a copy of an entity with normalized metric.

Point ProjectPoint(Point point)
Returns the location on the geom that is closest to given point.

Geom Segmentize(Number length)
Returns a copy of an entity with no segment longer than specified value.

GeomSet SplitWithLines(GeomSet geomSet, Number epsilon)
Returns set of geometric entities composed from subject entity by splitting it with given set of lines.

Geom Subtract(GeomSet geomSet, Number epsilon)
Returns a copy of an entity minus given set of entities.

Object ToBinary()
Creates a binary representation of a geom in Geometry format.

Object ToBinarySDE(Number shiftX, Number shiftY, Number scale)

Creates a binary representation of a geom in Geometry (SDE) format. Shift and scale parameters specify conversion factors to integer coordinates and are optional. If omitted, shiftX is set to 0, shiftY is set to 0, and scale is set to 1.

Object ToBinarySHP()

Creates a binary representation of a geom in Geometry (SHP) format.

Object ToBinaryWKB()

Creates a binary representation of a geom in Geometry (WKB) format.

String ToTextWKT()

Creates a textual representation of a geom in OGC WKT format.

Obtained From

Application (NewGeom)

Geom (Buffer, EnclosingRect, Intersect, Normalize, Segmentize, Subtract)
GeomSet (ConvexHull, EnclosingRect, FarthestPair, Item, LastAdded, ShapeHull, Union)
Object (Geom)

Notes

It is possible to make a Geom object created using a script invalid, as long as the object does not represent the metric of any real drawing object or label. For example, it is possible to delete the last point in a branch or delete the last branch of such a Geom object. This corrects the strange situation that was possible in releases before 6.50 SP1 when a standalone Geom object was invalid at creation time, but could no longer become invalid after it has been made valid.

See Also

Scripting Reference
**GeomSet Object**
Set of geometric entities.

**Properties**

Application Application
Returns application. Read only.

Rect Box
Sets or returns bounding box. Modifying this property scales and shifts the set. Read/write.

Point Center
Sets or returns center of the centroid of the set. Modifying this property moves the set. Read/write.

Number Count
Returns number of entities in set. Read only.

Geom Item(Number index)
Returns entity with given index. Read only.

Geom LastAdded
Returns last added entity if any. Read only.

Any Parent
Returns parent object. Read only.

**Methods**

Add(Geom geom)
Append entity to set.

GeomSet BoundedAreas(Number epsilon)
Returns areas bounded by a set of lines.

GeomSet Cluster(Number factor)
Returns lines connecting clusters of a set of points.

GeomSet ClusterZahn(Number factor)
Returns lines connecting Zahn clusters of a set of points.

GeomSet ConstrainedTriangulationEdges(Number epsilon)
Returns edges (line objects) of constrained Delaunay triangulation of a set of objects.

GeomSet ConstrainedTriangulationTiles(Number epsilon)
Returns tiles (area objects) of constrained Delaunay triangulation of a set of objects.

Geom ConvexHull(Number epsilon)
Returns convex hull of a set.

GeomSet DistanceNetwork(Number distance)
Returns lines connecting pairs of points from subject set that are closer to each other than given distance.
Number EnclosingCircle(Point center)
Returns radius of minimum enclosing circle and puts center of this circle into supplied parameter.

Geom EnclosingRect()
Returns minimum enclosing rectangle.

GeomSet FarthestNeighbor(GeomSet geomSet, Number epsilon)
Returns set of lines connecting points from subject set and their farthest neighbors from given set.

GeomSet FarthestNeighborSymm(GeomSet geomSet, Number epsilon)
Returns set of lines connecting points from subject set and their symmetric farthest neighbors from given set.

Geom FarthestPair(Number epsilon)
Returns line between farthest pair of points in a point set.

GeomSet GabrielNetwork()
Returns lines connecting Gabriel network of a set of points.

GeomSet IntersectLines(Number epsilon)
Returns a copy of set with resolved line intersections.

GeomSet IntersectionPoints(Number epsilon)
Returns all intersections of a set of lines.

GeomSet JoinLines(Number epsilon)
Returns a copy of set joining lines.

GeomSet NearestNeighbor(GeomSet geomSet, Number epsilon)
Returns set of lines connecting points from subject set and their nearest neighbors from given set.

GeomSet NearestNeighborSymm(GeomSet geomSet, Number epsilon)
Returns set of lines connecting points from subject set and their symmetric nearest neighbors from given set.

Geom NearestPair(Number epsilon)
Returns line between nearest pair of points in a point set.

GeomSet NormalizeTopology(Number epsilon)
Returns a copy of a set with normalized topological relations.

GeomSet RelativeNeighborhoodNetwork()
Returns lines connecting relative neighborhood network of a set of points.

Remove(Number index)
Remove entity with given index.

RemoveAll()
Removes all entities.

Geom ShapeHull(Number cells, Number epsilon)
Returns shape hull of a set computed over a grid with given number of cells along the shortest axis.

GeomSet SpanningTree()
Returns lines connecting minimum spanning tree of a set of points.
GeomSet TriangulationEdges(Number epsilon)
Returns edges of Delaunay triangulation of a set of points.

GeomSet TriangulationTiles(Number epsilon)
Returns tiles of Delaunay triangulation of a set of points.

GeomSet VoronoiEdges(Number epsilon)
Returns edges of Voronoi diagram of a set of points.

GeomSet VoronoiTiles(Number epsilon)
Returns tiles of Voronoi diagram of a set of points.

GeomSet VoronoiVertices(Number epsilon)
Returns vertices of Voronoi diagram of a set of points.

Geom Union(Number epsilon)
Returns a union of all areas within set.

Obtained From
Geom (Decompose, DecomposeConvex, SplitWithLines)

See Also
Scripting Reference
**History Object**
History log.

**Properties**

- **Application Application**
  Returns application. Read only.

- **Any Parent**
  Returns parent object. Read only.

**Methods**

- **Log(String text, Boolean popup)**
  Append text to log. Popup parameter is optional. When it is set to True, history window automatically pops up on output.

**Obtained From**

- Application (History)

**See Also**

Scripting Reference
Image Object
Image component.

Properties

Application Application
Returns application. Read only.

Color BackgroundColor
Sets or returns background color. Read / write.

Rect Box
Returns bounding box of all pixels. Read only.

Boolean Cached
Sets or returns caching option for shared component in Enterprise Edition. Read/write.

Number Channels
Return number of image channels. Read only.

ControlPointSet ControlPointSet
Returns set of control points. Read only.

CoordinateSystem CoordinateSystem
Sets or returns coordinate system. Read/write.

Boolean CoordinateSystemVerified
Checks whether or not the component coordinate system has been verified by the user, or alters the verification state. Read / write.

View DefaultView
Returns default view. Read only.

String Description
Sets or returns component description. Read/write.

Document Document
Returns owner document. Read only.

Folder Folder
Sets or returns containing folder. Read/write.

Number Height
Returns image height in pixels. Read only.

Number ID
Returns unique component ID. Read only.

Boolean IsCheckedOut
Checks if component is shared and checked out in Enterprise Edition. Read only.

Boolean IsCheckedOutRemotely
Checks if component is shared and checked out by another user in Enterprise Edition. Read only.
Boolean IsLinked()
  Returns True if image is linked and False otherwise.

Boolean IsModified
  Checks if component is modified since last save in Enterprise Edition. Read only.

Boolean IsOutdated
  Checks if component is shared and there is a later version of the component on the Enterprise server in Enterprise Edition. Read only.

Boolean IsReadOnly
  Checks if component is readonly (eg, shared and not checked out) in Enterprise Edition. Read only.

Boolean IsShared
  Checks if component is shared on an Enterprise server in Enterprise Edition. Read only.

Boolean IsWritable
  Checks if component is writable in Enterprise Edition. Read only.

Number LineRest
  Returns number of bytes used to pad each image line. Read only.

Number LineSize
  Returns number of bytes used to store each image line. Read only.

String LinkRowset
  Returns the source rowset of a linked component. Read only.

String LinkSource
  Returns the data source of a linked component. Read only.

String LinkTechnology
  Returns the data access technology used by a linked component. Read only.

String Name
  Sets or returns component name. Read/write.

String Note
  Sets or returns notes associated with the component. Read/write.

ComponentSet OwnedComponentSet
  Returns set of components bound to this component. Read only.

ComponentSet OwnedLayoutSet
  Returns set of layouts bound to this component. Read only.

Palette OwnedPalette
  Returns palette bound to this component. Read only.

Component Owner
  Returns owner component if any. Read only.
Any Parent  
Returns parent object. Read only.

PixelSet PixelSet  
Returns set of image pixels. Read only.

Number RemoteVersion  
Returns version of shared component available on the Enterprise server in Enterprise Edition. Read only.

ImageSelectionSet SavedSelectionSet  
Returns set of saved selections. Read only.

PixelSet Selection  
Returns set of selected pixels. Read only.

String Server  
Returns server identification string for shared component in Enterprise Edition. Read only.

String ShareStatus  
Returns textual report on component sharing status in Enterprise Edition. Read only.

ComponentType Type  
Returns component type. Read only.

String TypeName  
Returns textual representation of the component type. Read only.

Unlink()  
Unlinks component. If the component is not linked, does nothing.

Number Version  
Returns version of shared component within the local project in Enterprise Edition. Read only.

ViewSet ViewSet  
Returns set of views associated with this component. Read only.

Number Width  
Returns image width in pixels. Read only.

Number ZoomMax  
Sets or returns maximum zoom level. Read/write.

Number ZoomMin  
Sets or returns minimum zoom level. Read/write.

Methods

CheckIn()  
Checks in changes made to a shared component to the Enterprise server in Enterprise Edition.

CheckOut()  
Checks out a shared component for editing in Enterprise Edition.
Clear(Boolean selection)
Clears entire component or selection within the component. Calling the method without parameters clears entire component.

Copy(Boolean selection)
Copies entire component or selection within the component into the Clipboard. Calling the method without parameters copies entire component.

CopyBitmap(Boolean selection)
Copies entire component or selection within the component into the Clipboard as a bitmap. Calling the method without parameters copies entire component.

Cut(Boolean selection)
Copies entire component or selection within the component into the Clipboard and then clears it. Calling the method without parameters cuts entire component.

CutBitmap(Boolean selection)
Copies entire component or selection within the component into the Clipboard as a bitmap and then clears it. Calling the method without parameters cuts entire component.

GetLatestVersion()
Gets the latest version of a shared component from the Enterprise server in Enterprise Edition.

Boolean IsAlpha()
Returns True if image is RGBA and False otherwise.

Boolean IsColor()
Returns True if image is RGB and False otherwise.

Boolean IsCompressed()
Returns True if image is compressed and False otherwise.

Boolean IsDatabase()
Returns True if image is linked from a database and False otherwise.

Boolean IsLibrary()
Returns True if image is an image library and False otherwise. Read only.

Boolean IsManifold()
Returns True if image is linked from a Manifold Image Server and False otherwise.

Boolean IsMono()
Returns True if image is grayscale and False otherwise.

Boolean IsOGC()
Returns True if image is linked from an OGC WMS server and False otherwise.

Boolean IsPalette()
Returns True if image is palette and False otherwise.

Boolean IsTerraServer()
Returns True if image is linked from TerraServer and False otherwise.

Open()
Opens component.

OpenInNewWindow()
Opens component in new window.

Paste(Boolean replace)
Pastes the contents of the Clipboard into the component replacing current selection or leaving it intact. Calling the method without parameters replaces selection.

Print(Boolean useDialog)
Prints component.

ProjectTo(CoordinateSystem system, Boolean adjustResolution)
Projects component to another coordinate system. If the second parameter is True or omitted, the system automatically adjusts the local scale parameters of the coordinate system to match the current resolution of the image.

RenderAreaTo(String name, Number width, Number height, Rect area, Boolean clean)
Renders component area into an image with given name, width and height. If last parameter is True, legend, north arrow and scale bar are not rendered even if they are on.

RenderTo(String name, Number width, Number height, Boolean clean)
Render component into an image with given name, width and height. If last parameter is True, legend, north arrow and scale bar are not rendered even if they are turned on.

SelectAll()
Selects all pixels.

SelectInverse()
Selects pixels that were not selected and vice versa.

SelectNone()
Unselects all pixels.

UndoCheckOut()
Undoes changes made to a shared component and gets the latest version of the component from the Enterprise server in Enterprise Edition.

Obtained From
Document (NewImage)

See Also
Scripting Reference
**ImageSelectionSet Object**
Set of saved selections within image.

**Properties**

- **Application Application**
  Returns application. Read only.

- **Number Count**
  Returns number of selections in set. Read only.

- **PixelSet Item(Number index)**
  Returns selection with given index. Read only.

- **Any Parent**
  Returns parent object. Read only.

**Methods**

- **Number ItemByMask(Number mask)**
  Returns index of selection with given mask value or -1.

- **Number ItemByName(String name)**
  Returns index of selection with given name or -1.

**Obtained From**

- **Image (SavedSelectionSet)**

**See Also**

Scripting Reference
Import Object
Generic importer.

Properties

Application Application
Returns application. Read only.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string (eg. "Text Files (*.txt)"). Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of import properties. Read only.

Methods

Boolean CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String fileName, ConvertPrompt prompt)
Imports given file.

Obtained From

Application (NewImport)

See Also

Scripting Reference
ImportAdf Object
ADF importer.

Properties

Application Application
Returns application. Read only.

ConverterItemSet Columns
Returns set of import columns (vector data only). Read only.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

ConvertPolicy ConvertPolicy
Sets or returns column import policy (vector data only). Read/write.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

Boolean ImportAreas
Sets or returns area composition option (vector data) or area import option (TINs). Read/write.

Boolean ImportPoints
Sets or returns point import option (TINs only). Read/write.

Boolean ImportSurface
Sets or returns surface import option (TINs only). Read/write.

Boolean ImportTicks
Sets or returns tick import option (vector data only). Read/write.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of import properties. Read only.

Number SurfaceHeight
Sets or returns surface height (TINs only). Read/write.

Number SurfaceWidth
Sets or returns surface width (TINs only). Read/write.

Methods
Boolean CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String fileName, ConvertPrompt prompt)
Imports given file.

Obtained From

Application (NewImport("ADF"))

See Also

Scripting Reference
**ImportAdoNet Object**

Generic ADO.NET importer.

**Properties**

- **Application Application**
  Returns application. Read only.

- **ConverterItemSet Columns**
  Returns set of import columns. Read only.

- **ComponentSet ComponentSet**
  Returns set of components created by the import. Read only.

- **ConnectionString**
  Sets or returns connection string. Read/write.

- **ConvertPolicy ConvertPolicy**
  Sets or returns table import policy. Read/write.

- **DefaultExtension**
  Returns default file extension. Read only.

- **Filter**
  Returns filter string. Read only.

- **Name**
  Returns file format name. Read only.

- **Parent**
  Returns parent object. Read only.

- **PropertySet PropertySet**
  Returns set of import properties. Read only.

**Methods**

- **CanImport(ComponentType type)**
  Returns True if importer can read components of given type and False otherwise.

- **Import(String table, ConvertPrompt prompt)**
  Imports table from the data source specified by the connection string.

**Obtained From**

- **Application (NewImport("ADO.NET"))**

**See Also**

- Scripting Reference
ImportAdoNetOdbc Object
ODBC ADO.NET importer.

Properties

Application Application
Returns application. Read only.

ConverterItemSet Columns
Returns set of import columns. Read only.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

String ConnectionString
Sets or returns connection string. Read/write.

ConvertPolicy ConvertPolicy
Sets or returns table import policy. Read/write.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of import properties. Read only.

Methods

Boolean CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String table, ConvertPrompt prompt)
Imports table from the data source specified by the connection string.

Obtained From

Application (NewImport("ADO.NET ODBC"))

See Also

Scripting Reference
ImportAdoNetOleDb Object
OLE DB ADO.NET importer.

Properties

Application Application
Returns application. Read only.

ConverterItemSet Columns
Returns set of import columns. Read only.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

String ConnectionString
Sets or returns connection string. Read/write.

ConvertPolicy ConvertPolicy
Sets or returns table import policy. Read/write.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of import properties. Read only.

Methods

Boolean CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String table, ConvertPrompt prompt)
Imports table from the data source specified by the connection string.

Obtained From

Application (NewImport("ADO.NET OLE DB"))

See Also

Scripting Reference
ImportAdoNetOracle Object
Oracle ADO.NET importer.

Properties

Application Application
Returns application. Read only.

ConverterItemSet Columns
Returns set of import columns. Read only.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

String ConnectionString
Sets or returns connection string. Read/write.

ConvertPolicy ConvertPolicy
Sets or returns table import policy. Read/write.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of import properties. Read only.

Methods

String BuildConnectionString(String server, String login, String password)
Composes a connection string for Oracle ADO.NET provider. The last two arguments are optional.

Boolean CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String table, ConvertPrompt prompt)
Imports table from the data source specified by the connection string.

Obtained From

Application (NewImport("ADO.NET Oracle"))

See Also
**ImportAdoNetSqlServer Object**
SQL Server ADO.NET importer.

**Properties**

Application Application
   Returns application. Read only.

ConverterItemSet Columns
   Returns set of import columns. Read only.

ComponentSet ComponentSet
   Returns set of components created by the import. Read only.

ConnectionString
   Sets or returns connection string. Read/write.

ConvertPolicy ConvertPolicy
   Sets or returns table import policy. Read/write.

DefaultExtension
   Returns default file extension. Read only.

Filter
   Returns filter string. Read only.

Name
   Returns file format name. Read only.

Parent
   Returns parent object. Read only.

PropertySet PropertySet
   Returns set of import properties. Read only.

**Methods**

BuildConnectionString(String server, String login, String password, String database)
   Composes a connection string for SQL Server ADO.NET provider. The last three arguments are optional.

CanImport(ComponentType type)
   Returns True if importer can read components of given type and False otherwise.

Import(String table, ConvertPrompt prompt)
   Imports table from the data source specified by the connection string.

**Obtained From**

Application (NewImport("ADO.NET SQL Server"))

**See Also**
**ImportAdrg Object**
ADRG importer.

**Properties**

- **Application Application**
  Returns application. Read only.

- **ComponentSet ComponentSet**
  Returns set of components created by the import. Read only.

- **ConvertPolicy ConvertPolicy**
  Sets or returns module import policy. Read/write.

- **String DefaultExtension**
  Returns default file extension. Read only.

- **String Filter**
  Returns filter string. Read only.

- **ConverterItemSet Modules**
  Returns set of import modules. Read only.

- **String Name**
  Returns file format name. Read only.

- **Any Parent**
  Returns parent object. Read only.

- **ConverterPropertySet PropertySet**
  Returns set of import properties. Read only.

**Methods**

- **Boolean CanImport(ComponentType type)**
  Returns True if importer can read components of given type and False otherwise.

- **Import(String fileName, ConvertPrompt prompt)**
  Imports given file.

**Obtained From**

- **Application (NewImport("ADRG"))**

**See Also**

- **Scripting Reference**
**ImportAvhrr Object**
AVHRR importer.

**Properties**

Application Application  
Returns application.  Read only.

ComponentSet ComponentSet  
Returns set of components created by the import. Read only.

ConvertPolicy ConvertPolicy  
Sets or returns module import policy.  Read/write.

String DefaultExtension  
Returns default file extension.  Read only.

String Filter  
Returns filter string.  Read only.

ConverterItemSet Modules  
Returns set of import modules.  Read only.

String Name  
Returns file format name.  Read only.

Any Parent  
Returns parent object.  Read only.

ConverterPropertySet PropertySet  
Returns set of import properties.  Read only.

**Methods**

Boolean CanImport(ComponentType type)  
Returns True if importer can read components of given type and False otherwise.

Import(String fileName, ConvertPrompt prompt)  
Imports given file.

**Obtained From**

Application (NewImport("AVHRR"))

**See Also**

Scripting Reference
ImportBil Object

BIL importer.

Properties

Application Application
Returns application. Read only.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

ConvertPolicy ConvertPolicy
Sets or returns module import policy. Read/write.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

ConverterItemSet Modules
Returns set of import modules. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of import properties. Read only.

Methods

Boolean CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String fileName, ConvertPrompt prompt)
Imports given file.

Obtained From

Application (NewImport("BIL"))

See Also

Scripting Reference
**ImportBmp Object**

BMP importer.

**Properties**

**Application Application**

Returns application. Read only.

**ComponentSet ComponentSet**

Returns set of components created by the import. Read only.

**String DefaultExtension**

Returns default file extension. Read only.

**String Filter**

Returns filter string. Read only.

**String Name**

Returns file format name. Read only.

**Any Parent**

Returns parent object. Read only.

**ConverterPropertySet PropertySet**

Returns set of import properties. Read only.

**Methods**

**Boolean CanImport(ComponentType type)**

Returns True if importer can read components of given type and False otherwise.

**Import(String fileName, ConvertPrompt prompt)**

Imports given file.

**Obtained From**

Application (NewImport("BMP"))

**See Also**

Scripting Reference
ImportBna Object
BNA importer.

Properties

Application Application
Returns application. Read only.

ConverterItemSet Columns
Returns set of import columns. Read only.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

ConvertPolicy ConvertPolicy
Sets or returns column import policy. Read/write.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of import properties. Read only.

Methods

Boolean CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String fileName, ConvertPrompt prompt)
Imports given file.

Obtained From

Application (NewImport("BNA"))

See Also

Scripting Reference
**ImportCadrgCib Object**

CADRG/CIB importer.

**Properties**

- **Application**
  Returns application. Read only.

- **ComponentSet**
  Returns set of components created by the import. Read only.

- **ConvertPolicy**
  Sets or returns module import policy. Read/write.

- **DefaultExtension**
  Returns default file extension. Read only.

- **Filter**
  Returns filter string. Read only.

- **Modules**
  Returns set of import modules. Read only.

- **Name**
  Returns file format name. Read only.

- **Parent**
  Returns parent object. Read only.

- **PropertySet**
  Returns set of import properties. Read only.

**Methods**

- **CanImport(ComponentType type)**
  Returns True if importer can read components of given type and False otherwise.

- **Import(String fileName, ConvertPrompt prompt)**
  Imports given file.

**Obtained From**

- Application (NewImport("CADRG / CIB"))

**See Also**

Scripting Reference
ImportCeosSeaWifs Object
CEOS SeaWiFS importer.

Properties

Application Application
Returns application. Read only.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

ConvertPolicy ConvertPolicy
Sets or returns module import policy. Read/write.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

ConverterItemSet Modules
Returns set of import modules. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of import properties. Read only.

Methods

Boolean CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String fileName, ConvertPrompt prompt)
Imports given file.

Obtained From

Application (NewImport("CEOS SeaWiFS"))

See Also

Scripting Reference
ImportCsv Object
CSV importer.

Properties

Application Application
Returns application. Read only.

Boolean ColumnHeaderLine
Sets or returns column header line option. If True, the first line in the imported file contains column names. Read/write.

ConverterItemSet Columns
Returns set of import columns. Read only.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

String ConnectionString
Sets or returns connection string (path to a file). Read/write.

ConvertPolicy ConvertPolicy
Sets or returns column import policy. Read/write.

String DefaultExtension
Returns default file extension. Read only.

String Delimiter
Sets or returns the list delimiter character. Read/write.

String DelimiterDecimal
Sets or returns the decimal delimiter character. Read/write.

String Filter
Returns filter string. Read only.

Boolean ForceAnsiCharset
Sets or returns force ANSI character set option. If True, the import uses the default ANSI character set. If False, the import uses the default OEM character set. Read/write.

Boolean ForceText
Sets or returns force text types option. If True, all columns are imported as text. If False, the system determines the type of each column from the first few lines of the imported file. Read/write.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of import properties. Read only.
Number ScanLines
Sets or returns the number of lines to scan to determine column types. Has no effect if the ForceText property is set to True. Read/write.

String TextQualifier
Sets or returns the text qualifier character. Read/write.

Methods

Boolean CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String table, ConvertPrompt prompt)
Imports table from the data source specified by the connection string.

Obtained From

Application (NewImport("CSV"))

See Also

Scripting Reference
**ImportDb Object**

DB importer.

**Properties**

Application Application
Returns application. Read only.

ConverterItemSet Columns
Returns set of import columns. Read only.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

String ConnectionString
Sets or returns connection string (path to a file). Read/write.

ConvertPolicy ConvertPolicy
Sets or returns column import policy. Read/write.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of import properties. Read only.

**Methods**

Boolean CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String table, ConvertPrompt prompt)
Imports table from the data source specified by the connection string.

**Obtained From**

Application (NewImport("DB"))

**See Also**

Scripting Reference
ImportDb2 Object
Native DB2 import.

Properties

Application Application
Returns application. Read only.

ConverterItemSet Columns
Returns set of import columns. Read only.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

String ConnectionString
Sets or returns connection string. Read/write.

ConvertPolicy ConvertPolicy
Sets or returns table import policy. Read/write.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of import properties. Read only.

Methods

String BuildConnectionString(String server, String login, String password)
Composes a connection string for DB2 native client. The last two arguments are optional.

Boolean CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String table, ConvertPrompt prompt)
Imports table from the data source specified by the connection string.

Obtained From

Application (NewImport("DB2"))

See Also
Scripting Reference
**ImportDbf Object**

DBF importer.

**Properties**

*Application Application*
Returns application. Read only.

*ConverterItemSet Columns*
Returns set of import columns. Read only.

*ComponentSet ComponentSet*
Returns set of components created by the import. Read only.

*ConnectionString*
Sets or returns connection string (path to a file). Read/write.

*ConvertPolicy ConvertPolicy*
Sets or returns column import policy. Read/write.

*DefaultExtension*
Returns default file extension. Read only.

*Filter*
Returns filter string. Read only.

*Name*
Returns file format name. Read only.

*Parent*
Returns parent object. Read only.

*PropertySet PropertySet*
Returns set of import properties. Read only.

**Methods**

*CanImport(ComponentType type)*
Returns True if importer can read components of given type and False otherwise.

*Import(String table, ConvertPrompt prompt)*
Imports table from the data source specified by the connection string.

**Obtained From**

*Application (NewImport("DBF"))*

**See Also**

Scripting Reference
ImportDem Object
DEM importer.

Properties

Application Application
Returns application. Read only.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of import properties. Read only.

Methods

Boolean CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String fileName, ConvertPrompt prompt)
Imports given file.

Obtained From

Application (NewImport("DEM"))

See Also

Scripting Reference
ImportDemGlobe Object
DEM GLOBE importer.

Properties

Application Application
Returns application. Read only.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of import properties. Read only.

Methods

Boolean CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String fileName, ConvertPrompt prompt)
Imports given file.

Obtained From

Application (NewImport("DEM GLOBE"))

See Also

Scripting Reference
**ImportDemGtopo30 Object**
DEM GTOPO30 importer.

**Properties**

Application Application
Returns application. Read only.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of import properties. Read only.

**Methods**

Boolean CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String fileName, ConvertPrompt prompt)
Imports given file.

**Obtained From**

Application (NewImport("DEM GTOPO30"))

**See Also**

Scripting Reference
**ImportDgn Object**
DGN importer.

**Properties**

- **Application Application**
  Returns application. Read only.

- **ComponentSet ComponentSet**
  Returns set of components created by the import. Read only.

- **String DefaultExtension**
  Returns default file extension. Read only.

- **String Filter**
  Returns filter string. Read only.

- **Boolean ImportZ**
  Sets or returns height import option. Read/write.

- **String Name**
  Returns file format name. Read only.

- **Any Parent**
  Returns parent object. Read only.

- **ConverterPropertySet PropertySet**
  Returns set of import properties. Read only.

- **Number Threshold**
  Sets or returns approximation threshold for curved entities. Read/write.

**Methods**

- **Boolean CanImport(ComponentType type)**
  Returns True if importer can read components of given type and False otherwise.

- **Import(String fileName, ConvertPrompt prompt)**
  Imports given file.

**Obtained From**

Application (NewImport("DNG"))

**See Also**

Scripting Reference
ImportDlg Object
DLG importer.

Properties

Application Application
Returns application. Read only.

ConverterItemSet Columns
Returns set of import columns. Read only.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

ConvertPolicy ConvertPolicy
Sets or returns column import policy. Read/write.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

Boolean MakeAreas
Sets or returns area composition option. Read/write.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of import properties. Read only.

Methods

Boolean CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String fileName, ConvertPrompt prompt)
Imports given file.

Obtained From

Application (NewImport("DLG"))

See Also

Scripting Reference
**ImportDoq Object**

DOQ importer.

**Properties**

Application Application
Returns application. Read only.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of import properties. Read only.

**Methods**

Boolean CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String fileName, ConvertPrompt prompt)
Imports given file.

**Obtained From**

Application (NewImport("DOQ"))

**See Also**

Scripting Reference
ImportDsn Object
DSN importer.

Properties

Application Application
Returns application. Read only.

ConverterItemSet Columns
Returns set of import columns. Read only.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

String ConnectionString
Sets or returns connection string (path to a file). Read/write.

ConvertPolicy ConvertPolicy
Sets or returns table import policy. Read/write.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of import properties. Read only.

Methods

Boolean CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String table, ConvertPrompt prompt)
Imports table from the data source specified by the connection string.

Obtained From

Application (NewImport("DSN"))

See Also

Scripting Reference
ImportDted Object
DTED importer.

Properties

Application Application
Returns application. Read only.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of import properties. Read only.

Methods

Boolean CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String fileName, ConvertPrompt prompt)
Imports given file.

Obtained From

Application (NewImport("DTED"))

See Also

Scripting Reference
**ImportDwg Object**

DWG importer.

**Properties**

**Application**
Application
Returns application. Read only.

**ComponentSet**
ComponentSet
Returns set of components created by the import. Read only.

**String DefaultExtension**
Returns default file extension. Read only.

**String Filter**
Returns filter string. Read only.

**Boolean ImportZ**
Sets or returns height import option. Read/write.

**String Name**
Returns file format name. Read only.

**Any Parent**
Returns parent object. Read only.

**ConverterPropertySet PropertySet**
Returns set of import properties. Read only.

**Boolean StripRtfTags**
Returns or sets the value of the tag stripping mode for text labels. Read/write.

**Number Threshold**
Sets or returns approximation threshold for curved entities. Read/write.

**Boolean WidenLines**
Sets or returns the conversion of lines with non-zero widths to areas. Read/write.

**Methods**

**Boolean CanImport(ComponentType type)**
Returns True if importer can read components of given type and False otherwise.

**Import(String fileName, ConvertPrompt prompt)**
Imports given file.

**Obtained From**

Application (NewImport("DWG"))

**See Also**
**ImportDxf Object**

DXF importer.

**Properties**

- **Application Application**
  Returns application. Read only.

- **ComponentSet ComponentSet**
  Returns set of components created by the import. Read only.

- **String DefaultExtension**
  Returns default file extension. Read only.

- **String Filter**
  Returns filter string. Read only.

- **Boolean ImportZ**
  Sets or returns height import option. Read/write.

- **String Name**
  Returns file format name. Read only.

- **Any Parent**
  Returns parent object. Read only.

- **ConverterPropertySet PropertySet**
  Returns set of import properties. Read only.

- **Boolean StripRtfTags**
  Returns or sets the value of the tag stripping mode for text labels. Read / write.

- **Number Threshold**
  Sets or returns approximation threshold for curved entities. Read/write.

- **Boolean WidenLines**
  Sets or returns the conversion of lines with non-zero widths to areas. Read / write.

**Methods**

- **Boolean CanImport(ComponentType type)**
  Returns True if importer can read components of given type and False otherwise.

- **Import(String fileName, ConvertPrompt prompt)**
  Imports given file.

**Obtained From**

- **Application (NewImport("DXF"))**

**See Also**
Scripting Reference
**ImportE00 Object**

E00 importer.

**Properties**

- **Application Application**
  Returns application. Read only.

- **ConverterItemSet Columns**
  Returns set of import columns. Read only.

- **ComponentSet ComponentSet**
  Returns set of components created by the import. Read only.

- **ConvertPolicy ConvertPolicy**
  Sets or returns column import policy. Read/write.

- **String DefaultExtension**
  Returns default file extension. Read only.

- **String Filter**
  Returns filter string. Read only.

- **Boolean ImportAreas**
  Sets or returns area composition option. Read/write.

- **Boolean ImportTicks**
  Sets or returns tick import option. Read/write.

- **String Name**
  Returns file format name. Read only.

- **Any Parent**
  Returns parent object. Read only.

- **ConverterPropertySet PropertySet**
  Returns set of import properties. Read only.

**Methods**

- **Boolean CanImport(ComponentType type)**
  Returns True if importer can read components of given type and False otherwise.

- **Import(String fileName, ConvertPrompt prompt)**
  Imports given file.

**Obtained From**

- Application (NewImport("E00"))

**See Also**
Scripting Reference
ImportEcw Object
ECW importer.

Properties

Application Application
Returns application. Read only.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of import properties. Read only.

Methods

Boolean CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String fileName, ConvertPrompt prompt)
Imports given file.

Obtained From

Application (NewImport("ECW"))

See Also

Scripting Reference
**ImportEmf Object**

EMF importer.

**Properties**

**Application Application**
Returns application. Read only.

**ComponentSet ComponentSet**
Returns set of components created by the import. Read only.

**String DefaultExtension**
Returns default file extension. Read only.

**String Filter**
Returns filter string. Read only.

**Number Height**
Sets or returns height of imported image. Read/write.

**ImageType ImageType**
Sets or returns type of imported image (ImageTypeRGB or ImageTypeRGBA). Read/write.

**String Name**
Returns file format name. Read only.

**Any Parent**
Returns parent object. Read only.

**ConverterPropertySet PropertySet**
Returns set of import properties. Read only.

**Number Width**
Sets or returns width of imported image. Read/write.

**Methods**

**Boolean CanImport(ComponentType type)**
Returns True if importer can read components of given type and False otherwise.

**Import(String fileName, ConvertPrompt prompt)**
Imports given file.

**Obtained From**

Application (NewImport("EMF"))

**See Also**

Scripting Reference
**ImportEnvi Object**
ENVI importer.

**Properties**

*Application Application*
Returns application. Read only.

*ComponentSet ComponentSet*
Returns set of components created by the import. Read only.

*ConvertPolicy ConvertPolicy*
Sets or returns module import policy. Read / write.

*String DefaultExtension*
Returns default file extension. Read only.

*String Filter*
Returns filter string. Read only.

*ConverterItemSet Modules*
Returns set of import modules. Read only.

*String Name*
Returns file format name. Read only.

*Any Parent*
Returns parent object. Read only.

*ConverterPropertySet PropertySet*
Returns set of import properties. Read only.

*Subtype Any Subtype*
Sets or returns import subtype. Read / write.

**Methods**

*Boolean CanImport(ComponentType type)*
Returns True if importer can read components of given type and False otherwise.

*Import(String fileName, ConvertPrompt prompt)*
Imports given file.

**Obtained From**

*Application (NewImport("ENVI IMG"))*

**See Also**

Scripting Reference
**ImportEradas Object**

ERDAS importer.

**Properties**

Application Application
Returns application. Read only.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

ConvertPolicy ConvertPolicy
Sets or returns module import policy. Read/write.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

ConverterItemSet Modules
Returns set of import modules. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of import properties. Read only.

**Methods**

Boolean CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String fileName, ConvertPrompt prompt)
Imports given file.

**Obtained From**

Application (NewImport("ERDAS GIS"))

**See Also**

Scripting Reference
**ImportErdasImagine Object**

ERDAS Imagine importer.

**Properties**

Application Application
Returns application. Read only.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

ConvertPolicy ConvertPolicy
Sets or returns module import policy. Read/write.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

ConverterItemSet Modules
Returns set of import modules. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of import properties. Read only.

Subtype Any Subtype
Sets or returns import subtype. Read/write.

**Methods**

Boolean CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String fileName, ConvertPrompt prompt)
Imports given file.

**Obtained From**

Application (NewImport("ERDAS IMG"))

**See Also**

Scripting Reference
ImportErs Object
ERS importer.

Properties

Application Application
Returns application. Read only.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

ConvertPolicy ConvertPolicy
Sets or returns module import policy. Read/write.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

ConverterItemSet Modules
Returns set of import modules. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of import properties. Read only.

Subtype Any Subtype
Sets or returns import subtype. Read/write.

Methods

Boolean CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String fileName, ConvertPrompt prompt)
Imports given file.

Obtained From

Application (NewImport("ERS"))

See Also

Scripting Reference
ImportFlt Object
FLT importer.

**Properties**

- **Application Application**
  Returns application. Read only.

- **ComponentSet ComponentSet**
  Returns set of components created by the import. Read only.

- **String DefaultExtension**
  Returns default file extension. Read only.

- **String Filter**
  Returns filter string. Read only.

- **String Name**
  Returns file format name. Read only.

- **Any Parent**
  Returns parent object. Read only.

- **ConverterPropertySet PropertySet**
  Returns set of import properties. Read only.

**Methods**

- **Boolean CanImport(ComponentType type)**
  Returns True if importer can read components of given type and False otherwise.

- **Import(String fileName, ConvertPrompt prompt)**
  Imports given file.

**Obtained From**

- Application (NewImport("FLT"))

**See Also**

Scripting Reference
**ImportGcdb Object**
Manifold Geocoding Database importer.

**Properties**

Application Application
Returns application. Read only.

ConverterItemSet Columns
Returns set of import columns. Read only.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

ConvertPolicy ConvertPolicy
Sets or returns column import policy. Read/write.

String Counties
Sets or returns list of counties to import, separated by semicolons. Each county may include a state name, separated by a comma, for example, "San Luis Obispo, CA". Read/write.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

Boolean ImportFormatting
Sets or returns format import option. Read / write.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of import properties. Read only.

**Methods**

Boolean CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String fileName, ConvertPrompt prompt) Imports given file.

**Obtained From**

Application (NewImport("Geocoding Database"))

**See Also**

Scripting Reference
**ImportGdf Object**
GDF importer.

**Properties**

- **Application Application**
  Returns application. Read only.

- **ComponentSet ComponentSet**
  Returns set of components created by the import. Read only.

- **ConvertPolicy ConvertPolicy**
  Sets or returns column import policy. Read/write.

- **String DefaultExtension**
  Returns default file extension. Read only.

- **String Filter**
  Returns filter string. Read only.

- **Boolean ImportZ**
  Sets or returns height import option. Read/write.

- **Boolean MapColumns**
  Sets or returns column mapping option. Read/write.

- **ConverterItemSet Modules**
  Returns set of import modules. Read only.

- **String Name**
  Returns file format name. Read only.

- **Any Parent**
  Returns parent object. Read only.

- **ConverterPropertySet PropertySet**
  Returns set of import properties. Read only.

**Methods**

- **Boolean CanImport(ComponentType type)**
  Returns True if importer can read components of given type and False otherwise.

- **Import(String fileName, ConvertPrompt prompt)**
  Imports given file.

**Obtained From**

- **Application (NewImport("GDF"))**

**See Also**
ImportGeoSpot Object
GeoSPOT importer.

Properties

Application Application
Returns application. Read only.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

ConvertPolicy ConvertPolicy
Sets or returns module import policy. Read/write.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

ConverterItemSet Modules
Returns set of import modules. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of import properties. Read only.

Methods

Boolean CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String fileName, ConvertPrompt prompt)
Imports given file.

Obtained From

Application (NewImport("GeoSPOT"))

See Also

Scripting Reference
ImportGif Object
GIF importer.

Properties

Application Application
Returns application. Read only.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of import properties. Read only.

Methods

Boolean CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String fileName, ConvertPrompt prompt)
Imports given file.

Obtained From

Application (NewImport("GIF"))

See Also

Scripting Reference
**ImportGml Object**
GML importer. Current version is primarily oriented on OS MasterMap data.

**Properties**

* Application Application
  Returns application object. Read only.

* ComponentSet ComponentSet
  Returns set of components created by the import. Read only.

* String DefaultExtension
  Returns default file extension. Read only.

* String Filter
  Returns filter string. Read only.

* Boolean MasterMap
  Sets or returns the use of MasterMap-specific options. Read/write.

* String Name
  Returns file format name. Read only.

* Any Parent
  Returns parent object. Read only.

* ConverterPropertySet PropertySet
  Returns set of import properties. Read only.

* ConverterItemSet Updates
  Returns set of update modules. Read only.

**Methods**

* Boolean CanImport(ComponentType type)
  Returns True if importer can read components of given type and False otherwise.

* Import(String fileName, ConvertPrompt prompt)
  Imports given file.

**Obtained From**

Application (NewImport("GML"))

**See Also**

Scripting Reference
ImportGrass Object
GRASS importer.

Properties

Application Application
Returns application. Read only.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of import properties. Read only.

Methods

Boolean CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String fileName, ConvertPrompt prompt)
Imports given file.

Obtained From

Application (NewImport("GRASS"))

See Also

Scripting Reference
**ImportGrd Object**
ESRI ASCII grid importer.

**Properties**

- **Application Application**
  Returns application. Read only.

- **ComponentSet ComponentSet**
  Returns set of components created by the import. Read only.

- **String DefaultExtension**
  Returns default file extension. Read only.

- **String Filter**
  Returns filter string. Read only.

- **String Name**
  Returns file format name. Read only.

- **Any Parent**
  Returns parent object. Read only.

- **ConverterPropertySet PropertySet**
  Returns set of import properties. Read only.

**Methods**

- **Boolean CanImport(ComponentType type)**
  Returns True if importer can read components of given type and False otherwise.

- **Import(String fileName, ConvertPrompt prompt)**
  Imports given file.

**Obtained From**

- Application (NewImport("GRD"))

**See Also**

Scripting Reference
ImportGrdSurfer Object
Surfer GRD importer (imports Surfer 6 GRD, Surfer 7 GRD and Surfer ASCII GRD formats).

Properties

Application Application
Returns application. Read only.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of import properties. Read only.

Methods

Boolean CanImport((ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String fileName, ConvertPrompt prompt)
Imports given file.

Obtained From

Application (NewImport("Surfer GRD"))

See Also

Scripting Reference
**ImportGxf Object**
GXF importer.

**Properties**

Application Application
Returns application. Read only.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of import properties. Read only.

**Methods**

Boolean CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String fileName, ConvertPrompt prompt)
Imports given file.

**Obtained From**

Application (NewImport("GXF"))

**See Also**

Scripting Reference
ImportHdf Object
HDF importer.

Properties

Application Application
Returns application. Read only.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

ConvertPolicy ConvertPolicy
Sets or returns module import policy. Read/write.

Boolean CreateFolders
Sets or returns folder creation option. Read/write.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

ConverterItemSet Modules
Returns set of import modules. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of import properties. Read only.

Methods

Boolean CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String fileName, ConvertPrompt prompt)
Imports given file.

Obtained From

Application (NewImport("HDF"))

See Also

Scripting Reference
**ImportHdfEos Object**
HDF EOS importer.

**Properties**

Application Application
Returns application. Read only.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

ConvertPolicy ConvertPolicy
Sets or returns module import policy. Read/write.

Boolean CreateFolders
Sets or returns folder creation option. Read/write.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

ConverterItemSet Modules
Returns set of import modules. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of import properties. Read only.

Subtype Any Subtype
Sets or returns import subtype. Read/write.

**Methods**

Boolean CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String fileName, ConvertPrompt prompt)
Imports given file.

**Obtained From**

Application (NewImport("HDF EOS"))

**See Also**
**ImportHdfSeaWifs Object**

HDF SeaWiFS importer.

**Properties**

- **Application**
  - Application
  - Returns application. Read only.

- **ComponentSet**
  - ComponentSet
  - Returns set of components created by the import. Read only.

- **ConvertPolicy**
  - ConvertPolicy
  - Sets or returns module import policy. Read/write.

- **DefaultExtension**
  - DefaultExtension
  - Returns default file extension. Read only.

- **Filter**
  - Filter
  - Returns filter string. Read only.

- **Modules**
  - ModuleSet
  - Returns set of import modules. Read only.

- **Name**
  - Name
  - Returns file format name. Read only.

- **Parent**
  - Parent
  - Returns parent object. Read only.

- **PropertySet**
  - PropertySet
  - Returns set of import properties. Read only.

**Methods**

- **CanImport(ComponentType type)**
  - Returns True if importer can read components of given type and False otherwise.

- **Import(String fileName, ConvertPrompt prompt)**
  - Imports given file.

** Obtained From**

- Application (NewImport("HDF SeaWiFS"))

**See Also**

Scripting Reference
ImportHtml Object
HTML importer.

Properties

Application Application
Returns application. Read only.

ConverterItemSet Columns
Returns set of import columns. Read only.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

ConnectionString ConnectionString
Sets or returns connection string (path to a file). Read/write.

ConvertPolicy ConvertPolicy
Sets or returns column import policy. Read/write.

DefaultExtension DefaultExtension
Returns default file extension. Read only.

Filter Filter
Returns filter string. Read only.

Name Name
Returns file format name. Read only.

Parent Parent
Returns parent object. Read only.

PropertySet PropertySet
Returns set of import properties. Read only.

Methods

CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String table, ConvertPrompt prompt)
Imports table from the data source specified by the connection string.

Obtained From

Application (NewImport("HTML"))

See Also

Scripting Reference
**ImportImdisp Object**
IMDISP importer.

**Properties**

- **Application Application**
  Returns application. Read only.

- **ComponentSet ComponentSet**
  Returns set of components created by the import. Read only.

- **String DefaultExtension**
  Returns default file extension. Read only.

- **String Filter**
  Returns filter string. Read only.

- **String Name**
  Returns file format name. Read only.

- **Any Parent**
  Returns parent object. Read only.

- **ConverterPropertySet PropertySet**
  Returns set of import properties. Read only.

**Methods**

- **Boolean CanImport(ComponentType type)**
  Returns True if importer can read components of given type and False otherwise.

- **Import(String fileName, ConvertPrompt prompt)**
  Imports given file.

**Obtained From**

- **Application (NewImport("IMDISP"))**

**See Also**

Scripting Reference
**ImportJpeg Object**

JPEG importer.

**Properties**

- **Application Application**
  Returns application. Read only.

- **ComponentSet ComponentSet**
  Returns set of components created by the import. Read only.

- **String DefaultExtension**
  Returns default file extension. Read only.

- **String Filter**
  Returns filter string. Read only.

- **String Name**
  Returns file format name. Read only.

- **Any Parent**
  Returns parent object. Read only.

- **ConverterPropertySet PropertySet**
  Returns set of import properties. Read only.

**Methods**

- **Boolean CanImport(ComponentType type)**
  Returns True if importer can read components of given type and False otherwise.

- **Import(String fileName, ConvertPrompt prompt)**
  Imports given file.

**Obtained From**

- Application (NewImport("JPEG"))

**See Also**

- Scripting Reference
ImportKml Object
KML importer.

Properties

Application Application
Returns application. Read only.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of import properties. Read only.

Methods

Boolean CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String fileName, ConvertPrompt prompt)
Imports given file.

Obtained From

Application (NewImport("KML"))

See Also

Scripting Reference
**ImportLas Object**

LAS importer.

**Properties**

- **Application Application**
  Returns application. Read only.

- **ComponentSet ComponentSet**
  Returns set of components created by the import. Read only.

- **ConvertPolicy ConvertPolicy**
  Sets or returns module import policy. Read/write.

- **String DefaultExtension**
  Returns default file extension. Read only.

- **String Filter**
  Returns filter string. Read only.

- **ConverterItemSet Modules**
  Returns set of import modules. Read only.

- **String Name**
  Returns file format name. Read only.

- **Any Parent**
  Returns parent object. Read only.

- **ConverterPropertySet PropertySet**
  Returns set of import properties. Read only.

**Methods**

- **Boolean CanImport(ComponentType type)**
  Returns True if importer can read components of given type and False otherwise.

- **Import(String fileName, ConvertPrompt prompt)**
  Imports given file.

**Obtained From**

- Application (NewImport("LAS"))

**See Also**

Scripting Reference
**ImportLulc Object**

LULC importer.

**Properties**

Application Application
Returns application. Read only.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

ConvertPolicy ConvertPolicy
Sets or returns module import policy. Read/write.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

ConverterItemSet Modules
Returns set of import modules. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of import properties. Read only.

**Methods**

Boolean CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String fileName, ConvertPrompt prompt)
Imports given file.

**Obtained From**

Application (NewImport("LULC"))

**See Also**

Scripting Reference
ImportLulcGiras Object
LULC GIRAS importer.

Properties

Application Application
Returns application. Read only.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of import properties. Read only.

Methods

Boolean CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String fileName, ConvertPrompt prompt)
Imports given file.

Obtained From

Application (NewImport("LULC GIRAS"))

See Also

Scripting Reference
**ImportMapBase Object**

ETAK MapBase importer.

**Properties**

- **Application Application**
  Returns application. Read only.

- **ComponentSet ComponentSet**
  Returns set of components created by the import. Read only.

- **ConvertPolicy ConvertPolicy**
  Sets or returns column import policy. Read/write.

- **String DefaultExtension**
  Returns default file extension. Read only.

- **String Filter**
  Returns filter string. Read only.

- **ConverterItemSet Modules**
  Returns set of import modules. Read only.

- **String Name**
  Returns file format name. Read only.

- **Any Parent**
  Returns parent object. Read only.

- **ConverterPropertySet PropertySet**
  Returns set of import properties. Read only.

**Methods**

- **Boolean CanImport(ComponentType type)**
  Returns True if importer can read components of given type and False otherwise.

- **Import(String fileName, ConvertPrompt prompt)**
  Imports given file.

**Obtained From**

- Application (NewImport("ETAK MapBase"))

**See Also**

Scripting Reference
ImportMdb Object
MDB importer.

Properties

Application Application
Returns application. Read only.

ConverterItemSet Columns
Returns set of import columns. Read only.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

ConnectionString
Sets or returns connection string (path to a file). Read/write.

ConvertPolicy ConvertPolicy
Sets or returns column import policy. Read/write.

DefaultExtension
Returns default file extension. Read only.

Filter
Returns filter string. Read only.

Name
Returns file format name. Read only.

Parent
Returns parent object. Read only.

PropertySet PropertySet
Returns set of import properties. Read only.

Methods

CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String table, ConvertPrompt prompt)
Imports table from the data source specified by the connection string.

Obtained From

Application (NewImport("MDB"))

See Also

Scripting Reference
**ImportMfd Object**
MFD importer.

**Properties**

Application Application
Returns application. Read only.

ConverterItemSet Columns
Returns set of import columns. Read only.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

ConvertPolicy ConvertPolicy
Sets or returns column import policy. Read/write.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

Boolean Layered
Sets or returns layer import option. Read/write.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of import properties. Read only.

**Methods**

Boolean CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String fileName, ConvertPrompt prompt)
Imports given file.

**Obtained From**

Application (NewImport("MFD"))

**See Also**

Scripting Reference
**ImportMif Object**
MIF importer.

**Properties**

*Application Application*  
Returns application. Read only.

*ConverterItemSet Columns*  
Returns set of import columns. Read only.

*ComponentSet ComponentSet*  
Returns set of components created by the import. Read only.

*ConvertPolicy ConvertPolicy*  
Sets or returns column import policy. Read/write.

*String DefaultExtension*  
Returns default file extension. Read only.

*String Filter*  
Returns filter string. Read only.

*Boolean ImportFormatting*  
Sets or returns format import option. Read/write.

*String Name*  
Returns file format name. Read only.

*Any Parent*  
Returns parent object. Read only.

*ConverterPropertySet PropertySet*  
Returns set of import properties. Read only.

**Methods**

*Boolean CanImport(ComponentType type)*  
Returns True if importer can read components of given type and False otherwise.

*Import(String fileName, ConvertPrompt prompt)*  
Imports given file.

**Obtained From**

*Application (NewImport("MIF"))*

**See Also**

Scripting Reference
**ImportMws Object**

MWS importer.

**Properties**

Application Application
Returns application. Read only.

ComponentSet ComponentSet
    Returns set of components created by the import. Read only.

String DefaultExtension
    Returns default file extension. Read only.

String Filter
    Returns filter string. Read only.

String Name
    Returns file format name. Read only.

Any Parent
    Returns parent object. Read only.

ConverterPropertySet PropertySet
    Returns set of import properties. Read only.

**Methods**

Boolean CanImport(ComponentType type)
    Returns True if importer can read components of given type and False otherwise.

Import(String fileName, ConvertPrompt prompt)
    Imports given file.

**Obtained From**

Application (NewImport("MWS"))

**See Also**

Scripting Reference
ImportNetCdf Object
NetCDF importer.

Properties

Application Application
Returns application. Read only.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

ConvertPolicy ConvertPolicy
Sets or returns module import policy. Read/write.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

ConverterItemSet Modules
Returns set of import modules. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of import properties. Read only.

Methods

Boolean CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String fileName, ConvertPrompt prompt)
Imports given file.

Obtained From

Application (NewImport("NetCDF"))

See Also

Scripting Reference
ImportNitf Object

NITF importer.

Properties

Application Application
Returns application object. Read only.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of import properties. Read only.

Methods

Boolean CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String fileName, ConvertPrompt prompt)
Imports given file.

Obtained From

Application (NewImport("NITF"))

See Also

Scripting Reference
**ImportNorthwood Object**
Northwood importer.

**Properties**

Application Application
Returns application. Read only.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

Boolean PlainZ
Returns or sets height conversion mode. False imports values as is. True will scale and offset values using coefficients stored within the file. Read/write.

ConverterPropertySet PropertySet
Returns set of import properties. Read only.

**Methods**

Boolean CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String fileName, ConvertPrompt prompt)
Imports given file.

**Obtained From**

Application (NewImport("Northwood GRC/GRD"))

**See Also**

Scripting Reference
ImportNtad Object
NTAD importer.

Properties

Application Application
Returns application. Read only.

ConverterItemSet Columns
Returns set of import columns. Read only.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

ConvertPolicy ConvertPolicy
Sets or returns column import policy. Read/write.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

Boolean MapColumns
Sets or returns column mapping option. Read/write.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of import properties. Read only.

Methods

Boolean CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String fileName, ConvertPrompt prompt)
Imports given file.

Obtained From

Application (NewImport("NTAD"))

See Also

Scripting Reference
ImportNtf Object
NTF importer.

Properties

Application Application
Returns application.  Read only.

ConverterItemSet Columns
Returns set of import columns.  Read only.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

ConvertPolicy ConvertPolicy
Sets or returns column import policy.  Read/write.

String DefaultExtension
Returns default file extension.  Read only.

String Filter
Returns filter string.  Read only.

String Name
Returns file format name.  Read only.

Any Parent
Returns parent object.  Read only.

ConverterPropertySet PropertySet
Returns set of import properties.  Read only.

Methods

Boolean CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String fileName, ConvertPrompt prompt)
Imports given file.

Obtained From

Application (NewImport("NTF"))

See Also

Scripting Reference
**ImportOdbc Object**
ODBC importer.

**Properties**

Application
Application
Returns application. Read only.

Columns
Returns set of import columns. Read only.

ComponentSet
Returns set of components created by the import. Read only.

ConnectionString
Sets or returns connection string. Read/write.

ConvertPolicy
Sets or returns table import policy. Read/write.

DefaultExtension
Returns default file extension. Read only.

Filter
Returns filter string. Read only.

Name
Returns file format name. Read only.

Parent
Returns parent object. Read only.

PropertySet
Returns set of import properties. Read only.

**Methods**

CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String table, ConvertPrompt prompt)
Imports table from the data source specified by the connection string.

**Obtained From**

Application (NewImport("ODBC"))

**See Also**

Scripting Reference
**ImportOleDb Object**

OLE DB importer.

**Properties**

Application Application
Returns application. Read only.

Columns
Returns set of import columns. Read only.

ComponentSet
Returns set of components created by the import. Read only.

ConnectionString
Sets or returns connection string. Read/write.

ConvertPolicy
Sets or returns column import policy. Read/write.

DefaultExtension
Returns default file extension. Read only.

Filter
Returns filter string. Read only.

Name
Returns file format name. Read only.

Parent
Returns parent object. Read only.

PropertySet
Returns set of import properties. Read only.

**Methods**

CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String table, ConvertPrompt prompt)
Imports table from the data source specified by the connection string.

**Obtained From**

Application (NewImport("OLEDB"))

**See Also**

Scripting Reference
ImportOracle Object
Native Oracle import.

Properties

Application Application
- Returns application. Read only.

ConverterItemSet Columns
- Returns set of import columns. Read only.

ComponentSet ComponentSet
- Returns set of components created by the import. Read only.

String ConnectionString
- Sets or returns connection string. Read/write.

ConvertPolicy ConvertPolicy
- Sets or returns table import policy. Read/write.

String DefaultExtension
- Returns default file extension. Read only.

String Filter
- Returns filter string. Read only.

String Name
- Returns file format name. Read only.

Any Parent
- Returns parent object. Read only.

ConverterPropertySet PropertySet
- Returns set of import properties. Read only.

Methods

String BuildConnectionString(String server, String login, String password)
- Composes a connection string for Oracle native client. The last two arguments are optional.

Boolean CanImport(ComponentType type)
- Returns True if importer can read components of given type and False otherwise.

Import(String table, ConvertPrompt prompt)
- Imports table from the data source specified by the connection string.

Obtained From

Application (NewImport("Oracle"))

See Also
Scripting Reference
ImportPcx Object
PCX importer.

Properties

Application Application
Returns application. Read only.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of import properties. Read only.

Methods

Boolean CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String fileName, ConvertPrompt prompt)
Imports given file.

Obtained From

Application (NewImport("PCX"))

See Also

Scripting Reference
**ImportPix Object**
PIX importer.

**Properties**

*Application Application*
Returns application. Read only.

*ComponentSet ComponentSet*
Returns set of components created by the import. Read only.

*ConvertPolicy ConvertPolicy*
Sets or returns module import policy. Read/write.

*String DefaultExtension*
Returns default file extension. Read only.

*String Filter*
Returns filter string. Read only.

*ConverterItemSet Modules*
Returns set of import modules. Read only.

*String Name*
Returns file format name. Read only.

*Any Parent*
Returns parent object. Read only.

*ConverterPropertySet PropertySet*
Returns set of import properties. Read only.

**Methods**

*Boolean CanImport(ComponentType type)*
Returns True if importer can read components of given type and False otherwise.

*Import(String fileName, ConvertPrompt prompt)*
Imports given file.

**Obtained From**

Application (NewImport("PIX"))

**See Also**

Scripting Reference
**ImportPng Object**
PNG importer.

**Properties**

Application Application
Returns application. Read only.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of import properties. Read only.

**Methods**

Boolean CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String fileName, ConvertPrompt prompt)
Imports given file.

**Obtained From**

Application (NewImport("PNG"))

**See Also**

Scripting Reference
ImportPostgreSQL Object
Native PostgreSQL import.

Properties

Application Application
Returns application. Read only.

Columns
Returns set of import columns. Read only.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

ConnectionString
Sets or returns connection string. Read/write.

ConvertPolicy ConvertPolicy
Sets or returns table import policy. Read/write.

DefaultExtension
Returns default file extension. Read only.

Filter
Returns filter string. Read only.

Name
Returns file format name. Read only.

Parent
Returns parent object. Read only.

PropertySet PropertySet
Returns set of import properties. Read only.

Methods

BuildConnectionString(String server, String login, String password, String database)
Composes a connection string for PostgreSQL native client. All arguments except server are optional.

CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String table, ConvertPrompt prompt)
Imports table from the data source specified by the connection string.

Obtained From

Application (NewImport("PostgreSQL")

See Also
**ImportPpm Object**
PPM importer.

**Properties**

- **Application Application**
  Returns application. Read only.

- **ComponentSet ComponentSet**
  Returns set of components created by the import. Read only.

- **String DefaultExtension**
  Returns default file extension. Read only.

- **String Filter**
  Returns filter string. Read only.

- **String Name**
  Returns file format name. Read only.

- **Any Parent**
  Returns parent object. Read only.

- **ConverterPropertySet PropertySet**
  Returns set of import properties. Read only.

**Methods**

- **Boolean CanImport(ComponentType type)**
  Returns True if importer can read components of given type and False otherwise.

- **Import(String fileName, ConvertPrompt prompt)**
  Imports given file.

**Obtained From**

Application (NewImport("PPM"))

**See Also**

Scripting Reference
ImportRawAscii Object

RAW ASCII importer.

Properties

Application Application
Returns application. Read only.

ChannelInterleaving ChannelInterleaving
Sets or returns channel interleaving option. Read/write.

Number Channels
Sets or returns number of channels. Read/write.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

String DefaultExtension
Returns default file extension. Read only.

String Delimiter
Sets or returns delimiter string. Read/write.

String Filter
Returns filter string. Read only.

Number HeaderLines
Sets or returns number of header lines to skip. Read/write.

Number Height
Sets or returns number of image lines. Read/write.

Number MissingValue
Sets or returns missing value. Read/write.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of import properties. Read only.

ValueType ValueType
Sets or returns type of pixel value. Read/write.

Number Width
Sets or returns number of pixels in image line. Read/write.

Methods
Boolean CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String fileName, ConvertPrompt prompt)
Imports given file.

**Obtained From**

Application (NewImport("Raw Text"))

**See Also**

Scripting Reference
**ImportRawBinary Object**

RAW binary importer.

**Properties**

Application Application
Returns application. Read only.

ByteOrder ByteOrder
Sets or returns byte order. Read/write.

ChannelInterleaving ChannelInterleaving
Sets or returns channel interleaving option. Read/write.

Number Channels
Sets or returns number of image channels. Read/write.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

Number Height
Sets or returns number of image lines. Read/write.

Number MissingValue
Sets or returns missing value. Read/write.

String Name
Returns file format name. Read only.

Number Offset
Sets or returns number of bytes to skip before reading data. Read/write.

Number Padding
Sets or returns number of bytes to skip after each line. Read/write.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of import properties. Read only.

ValueType ValueType
Sets or returns type of pixel value. Read/write.

Number Width
Sets or returns number of pixels in image line. Read/write.
Methods

Boolean CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String fileName, ConvertPrompt prompt)
Imports given file.

Obtained From

Application (NewImport("Raw Binary"))

See Also

Scripting Reference
**ImportRst Object**
RST importer.

**Properties**

Application Application
Returns application. Read only.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of import properties. Read only.

**Methods**

Boolean CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String fileName, ConvertPrompt prompt)
Imports given file.

**Obtained From**

Application (NewImport("IDRISI IMG"))

**See Also**

Scripting Reference
ImportS57 Object
S-57 importer.

Properties

Application Application
Returns application. Read only.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of import properties. Read only.

Methods

Boolean CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String fileName, ConvertPrompt prompt)
Imports given file.

Obtained From

Application (NewImport("S-57"))

See Also

Scripting Reference
**ImportSdts Object**
SDTS importer.

**Properties**

Application Application
Returns application. Read only.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

ConvertPolicy ConvertPolicy
Sets or returns column import policy. Read/write.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

Boolean ImportDataDefScheme
Sets or returns data definition scheme import option. Read/write.

Boolean ImportZ
Sets or returns height import option. Read/write.

Boolean MapColumns
Sets or returns column mapping option. Read/write.

ConverterItemSet Modules
Returns set of import modules. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of import properties. Read only.

**Methods**

Boolean CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String fileName, ConvertPrompt prompt)
Imports given file.

**Obtained From**

Application (NewImport("SDTS"))
See Also

Scripting Reference
**ImportSgi Object**
SGI importer.

**Properties**

Application Application
Returns application. Read only.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of import properties. Read only.

**Methods**

Boolean CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String fileName, ConvertPrompt prompt)
Imports given file.

**Obtained From**

Application (NewImport("SGI"))

**See Also**

Scripting Reference
**ImportShp Object**

SHP importer.

**Properties**

Application Application

Returns application. Read only.

ConverterItemSet Columns

Returns set of import columns. Read only.

ComponentSet ComponentSet

Returns set of components created by the import. Read only.

ConvertPolicy ConvertPolicy

Sets or returns column import policy. Read/write.

String DefaultExtension

Returns default file extension. Read only.

String Filter

Returns filter string. Read only.

Boolean ImportM

Sets or returns measure import option. Read/write.

Boolean ImportZ

Sets or returns height import option. Read/write.

String Name

Returns file format name. Read only.

Any Parent

Returns parent object. Read only.

ConverterPropertySet PropertySet

Returns set of import properties. Read only.

**Methods**

Boolean CanImport(ComponentType type)

Returns True if importer can read components of given type and False otherwise.

Import(String fileName, ConvertPrompt prompt)

Imports given file.

**Obtained From**

Application (NewImport("SHP"))

**See Also**
ImportSid Object
SID importer.

Properties

Application Application
Returns application. Read only.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

String Password
Sets or returns password for a password-protected file. Read/write.

ConverterPropertySet PropertySet
Returns set of import properties. Read only.

Number Scale
Sets or returns scale reduction factor. Read/write.

Methods

Boolean CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String fileName, ConvertPrompt prompt)
Imports given file.

Obtained From

Application (NewImport("SID"))

See Also

Scripting Reference
**ImportSpot Object**

SPOT importer.

**Properties**

- **Application Application**
  Returns application. Read only.

- **ComponentSet ComponentSet**
  Returns set of components created by the import. Read only.

- **ConvertPolicy ConvertPolicy**
  Sets or returns module import policy. Read/write.

- **String DefaultExtension**
  Returns default file extension. Read only.

- **String Filter**
  Returns filter string. Read only.

- **ConverterItemSet Modules**
  Returns set of import modules. Read only.

- **String Name**
  Returns file format name. Read only.

- **Any Parent**
  Returns parent object. Read only.

- **ConverterPropertySet PropertySet**
  Returns set of import properties. Read only.

**Methods**

- **Boolean CanImport(ComponentType type)**
  Returns True if importer can read components of given type and False otherwise.

- **Import(String fileName, ConvertPrompt prompt)**
  Imports given file.

**Obtained From**

- **Application (NewImport("SPOT"))**

**See Also**

- **Scripting Reference**
**ImportSqlServer Object**

SQL Server import.

**Properties**

- **Application**
  - Application
  - Returns application. Read only.

- **ConverterItemSet Columns**
  - Returns set of import columns. Read only.

- **ComponentSet ComponentSet**
  - Returns set of components created by the import. Read only.

- **ConnectionString**
  - Sets or returns connection string. Read/write.

- **ConvertPolicy ConvertPolicy**
  - Sets or returns table import policy. Read/write.

- **DefaultExtension**
  - Returns default file extension. Read only.

- **Filter**
  - Returns filter string. Read only.

- **Name**
  - Returns file format name. Read only.

- **Parent**
  - Returns parent object. Read only.

- **PropertySet PropertySet**
  - Returns set of import properties. Read only.

**Methods**

- **BuildConnectionString(String server, String login, String password, String database)**
  - Composes a connection string for SQL Server client. All arguments except server are optional.

- **CanImport(ComponentType type)**
  - Returns True if importer can read components of given type and False otherwise.

- **Import(String table, ConvertPrompt prompt)**
  - Imports table from the data source specified by the connection string.

**Obtained From**

- Application (NewImport("SQL Server"))

**See Also**
ImportSrtm Object
SRTM importer.

Properties

Application Application
Returns application object. Read only.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of import properties. Read only.

Methods

Boolean CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String fileName, ConvertPrompt prompt)
Imports given file.

Obtained From

Application (NewImport("SRTM"))

See Also

Scripting Reference
**ImportSun Object**

SUN importer.

**Properties**

- **Application**
  Returns application. Read only.

- **ComponentSet**
  Returns set of components created by the import. Read only.

- **DefaultExtension**
  Returns default file extension. Read only.

- **Filter**
  Returns filter string. Read only.

- **Name**
  Returns file format name. Read only.

- **Parent**
  Returns parent object. Read only.

- **PropertySet**
  Returns set of import properties. Read only.

**Methods**

- **CanImport(ComponentType type)**
  Returns True if importer can read components of given type and False otherwise.

- **Import(String fileName, ConvertPrompt prompt)**
  Imports given file.

**Obtained From**

Application (NewImport("SUN"))

**See Also**

Scripting Reference
ImportTab Object
TAB importer.

Properties

Application Application
Returns application. Read only.

ConverterItemSet Columns
Returns set of import columns. Read only.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

ConvertPolicy ConvertPolicy
Sets or returns column import policy. Read/write.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

Boolean ImportFormatting
Sets or returns format import option. Read/write.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of import properties. Read only.

Methods

Boolean CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String fileName, ConvertPrompt prompt)
Imports given file.

Obtained From

Application (NewImport("TAB"))

See Also

Scripting Reference
**ImportTaif Object**

TAIF importer.

**Properties**

Application Application
Returns application. Read only.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

ConvertPolicy ConvertPolicy
Sets or returns module import policy. Read/write.

DefaultExtension String
Returns default file extension. Read only.

Filter String
Returns filter string. Read only.

Modules ConverterItemSet
Returns set of import modules. Read only.

Name String
Returns file format name. Read only.

Parent Any
Returns parent object. Read only.

PropertySet ConverterPropertySet
Returns set of import properties. Read only.

Subtype Subtype
Sets or returns subtype of import file. Read/write.

**Methods**

CanImport(CompatibleType type)
Returns True if importer can read components of given type and False otherwise.

Import(String fileName, ConvertPrompt prompt)
Imports given file.

**Obtained From**

Application (NewImport("TAIF"))

**See Also**

Scripting Reference
ImportTga Object
TGA importer.

Properties

Application Application
Returns application. Read only.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of import properties. Read only.

Methods

Boolean CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String fileName, ConvertPrompt prompt)
Imports given file.

Obtained From

Application (NewImport("TGA"))

See Also

Scripting Reference
ImportTiff Object
TIFF importer.

Properties

Application Application
Returns application. Read only.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of import properties. Read only.

Methods

Boolean CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String fileName, ConvertPrompt prompt)
Imports given file.

Obtained From

Application (NewImport("TIFF"))

See Also

Scripting Reference
ImportTiger Object
TIGER/Line importer.

Properties

Application Application
Returns application. Read only.

ConverterItemSet Columns
Returns set of import columns. Read only.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

ConvertPolicy ConvertPolicy
Sets or returns column import policy. Read/write.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

Boolean MakeAreas
Sets or returns area composition option. Read/write.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of import properties. Read only.

Methods

Boolean CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String fileName, ConvertPrompt prompt)
Imports given file.

Obtained From

Application (NewImport("TIGER/Line"))

See Also

Scripting Reference
ImportTxt Object
TXT importer.

Properties

Application Application
Returns application. Read only.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

ConvertPolicy ConvertPolicy
Sets or returns table import policy. Read/write.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of import properties. Read only.

Methods

Boolean CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String fileName, ConvertPrompt prompt)
Imports given file.

Obtained From

Application (NewImport("TXT"))

See Also

Scripting Reference
ImportUdl Object
UDL importer.

Properties

Application Application
Returns application. Read only.

ConverterItemSet Columns
Returns set of import columns. Read only.

String ConnectionString
Sets or returns connection string (path to a file). Read/write.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

ConvertPolicy ConvertPolicy
Sets or returns table import policy. Read/write.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of import properties. Read only.

Methods

Boolean CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String table, ConvertPrompt prompt)
Imports table from the data source specified by the connection string.

Obtained From

Application (NewImport("UDL"))

See Also

Scripting Reference
**ImportVct Object**

VCT importer.

**Properties**

- **Application Application**
  Returns application. Read only.

- **ComponentSet ComponentSet**
  Returns set of components created by the import. Read only.

- **String DefaultExtension**
  Returns default file extension. Read only.

- **String Filter**
  Returns filter string. Read only.

- **String Name**
  Returns file format name. Read only.

- **Any Parent**
  Returns parent object. Read only.

- **ConverterPropertySet PropertySet**
  Returns set of import properties. Read only.

**Methods**

- **Boolean CanImport(ComponentType type)**
  Returns True if importer can read components of given type and False otherwise.

- **Import(String fileName, ConvertPrompt prompt)**
  Imports given file.

**Obtained From**

Application (NewImport("IDRISI VCT"))

**See Also**

Scripting Reference
**ImportVmap Object**

VMAP importer.

**Properties**

**Application Application**
Returns application. Read only.

**ComponentSet ComponentSet**
Returns set of components created by the import. Read only.

**ConvertPolicy ConvertPolicy**
Sets or returns module import policy. Read/write.

**Boolean CreateFolders**
Sets or returns folder creation option. Read/write.

**String DefaultExtension**
Returns default file extension. Read only.

**String Filter**
Returns filter string. Read only.

**Boolean ImportColumns**
Sets or returns column import option. Read/write.

**Boolean ImportZ**
Sets or returns height import option. Read/write.

**String Library**
Sets or returns library name. Read/write.

**ConverterItemSet Modules**
Returns set of import modules. Read only.

**String Name**
Returns file format name. Read only.

**Any Parent**
Returns parent object. Read only.

**ConverterPropertySet PropertySet**
Returns set of import properties. Read only.

**Methods**

**Boolean CanImport(ComponentType type)**
Returns True if importer can read components of given type and False otherwise.

**Import(String fileName, ConvertPrompt prompt)**
Imports given file.
Obtained From

Application (NewImport("VMAP"))

See Also

Scripting Reference
ImportWk Object
WK importer.

Properties

Application
Returns application. Read only.

ConverterItemSet
Returns set of import columns. Read only.

ConnectionString
Sets or returns connection string (path to a file). Read/write.

ComponentSet
Returns set of components created by the import. Read only.

ConvertPolicy
Sets or returns column import policy. Read/write.

DefaultExtension
Returns default file extension. Read only.

Filter
Returns filter string. Read only.

Name
Returns file format name. Read only.

Parent
Returns parent object. Read only.

ConverterPropertySet
Returns set of import properties. Read only.

Methods

CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String table, ConvertPrompt prompt)
Imports table from the data source specified by the connection string.

Obtained From

Application (NewImport("WK"))

See Also

Scripting Reference
**ImportXls Object**

XLS importer.

**Properties**

Application Application
Returns application. Read only.

ConverterItemSet Columns
Returns set of import columns. Read only.

String ConnectionString
Sets or returns connection string (path to a file). Read/write.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

ConvertPolicy ConvertPolicy
Sets or returns column import policy. Read/write.

String DefaultExtension
Returns default file extension. Read only.

String Filter
Returns filter string. Read only.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of import properties. Read only.

**Methods**

Boolean CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String table, ConvertPrompt prompt)
Imports table from the data source specified by the connection string.

**Obtained From**

Application (NewImport(“XLS”))

**See Also**

Scripting Reference
**ImportXyz Object**

XYZ importer.

**Properties**

Application Application
Returns application. Read only.

ComponentSet ComponentSet
Returns set of components created by the import. Read only.

String DefaultExtension
Returns default file extension. Read only.

String Delimiter
Sets or returns delimiter string. Read/write.

String Filter
Returns filter string. Read only.

Number HeaderLines
Sets or returns number of header lines to skip. Read/write.

Number Height
Sets or returns number of image lines. Read/write.

Number MissingValue
Sets or returns missing value. Read/write.

String Name
Returns file format name. Read only.

Any Parent
Returns parent object. Read only.

ConverterPropertySet PropertySet
Returns set of import properties. Read only.

ValueType ValueType
Sets or returns type of pixel value. Read/write.

Number Width
Sets or returns number of pixels in image line. Read/write.

**Methods**

Boolean CanImport(ComponentType type)
Returns True if importer can read components of given type and False otherwise.

Import(String fileName, ConvertPrompt prompt)
Imports given file.

Scan(String name)
Scans given file and autodetects import parameters.

**Obtained From**

Application (NewImport("XYZ"))

**See Also**

Scripting Reference
Label Object

Label.

Properties

Application Application
Returns application. Read only.

Color BackColor
Sets or returns background color. Read/write.

Boolean Bold
Returns True if the label font is bold and False otherwise, or modifies the font. Read/write.

Color BorderForeColor
Sets or returns border foreground color. Read/write.

Number CharSet
Sets or returns the character set used by the label font. Read/write.

String FontFace
Sets or returns the facename of the label font. Read/write.

Color ForeColor
Sets or returns foreground color. Read/write.

Geom Geom
Sets or returns label metric. Read/write.

Number ID
Returns unique label ID. Read only.

Boolean Italic
Returns True if the label font is italic and False otherwise, or modifies the font. Read/write.

LabelAlignMulti LabelAlignMulti
Sets or returns multiline label alignment. Read/write.

LabelAlignX LabelAlignX
Sets or returns horizontal label alignment. Read/write.

LabelAlignY LabelAlignY
Sets or returns vertical label alignment. Read/write.

Object Link
Returns owner object. Read only.

Number Mask
Sets or returns label mask. Read/write.

Component OwnerComponent
Returns labels component containing this label. Read only.
Any Parent
Returns parent object. Read only.

Number Rotation
Sets or returns label rotation in degrees. Read/write.

Boolean Selected
Selects or deselects label or checks if label is selected. Read/write.

Number Size
Sets or returns label text size in points. Read/write.

Boolean StrikeOut
Returns True if the label font is struck out and False otherwise, or modifies the font. Read/write.

Number Style
Sets or returns label style. Read/write.

String Text
Sets or returns label text. Read/write.

Boolean Underline
Returns True if the label font is underlined and False otherwise, or modifies the font. Read/write.

Boolean UsesCustomFormat
Returns True if the label uses custom format and False otherwise. Read only.

Methods

Boolean IsLine()
Returns True if label is aligned to line and False otherwise.

Boolean IsLinked()
Returns True if label if linked to object and False otherwise.

Boolean IsMasked(Number mask)
Returns True if label has given mask or masks and False otherwise.

Boolean IsPoint()
Returns True if label is aligned to point and False otherwise.

Boolean IsSelected()
Returns True if label is selected and False otherwise. This method is deprecated. Use the Selected property instead.

ResetFormat()
Resets label format to the default values.

Obtained From

EventArgs (Label)
LabelSet (Item, LastAdded)
LabelWindow (ActiveLabel)
MapWindow (ActiveLabel)

See Also

Scripting Reference
**Labels Object**
Labels component.

**Properties**

Application
Returns application. Read only.

BackgroundColor
Sets or returns background color. Read / write.

Boolean Bold
Returns True if component font is bold and False otherwise, or modifies the font. Read/write.

Rect Box
Returns bounding box of all labels. Read only.

Boolean Cached
Sets or returns caching option for shared component in Enterprise Edition. Read/write.

Number CalloutDistance
Sets or returns minimal length of a callout line, in points. Read/write.

Boolean Callouts
Returns True if the system displays a callout line for each point label and False otherwise, or turns callout lines on or off. Read/write.

Number CharSet
Sets or returns the character set used by component font. Read/write.

ControlPointSet
Returns set of control points. Read only.

CoordinateSystem
Sets or returns coordinate system. Read/write.

Boolean CoordinateSystemVerified
Checks whether or not the component coordinate system has been verified by the user, or alters the verification state. Read / write.

View
Returns default view. Read only.

String Description
Sets or returns component description. Read/write.

Document
Returns owner document. Read only.

Folder
Sets or returns containing folder. Read/write.

String FontFace
Sets or returns the facename of the component font. Read/write.

Number ID
Returns unique component ID. Read only.

Boolean IsCheckedOut
Checks if component is shared and checked out in Enterprise Edition. Read only.

Boolean IsCheckedOutRemotely
Checks if component is shared and checked out by another user in Enterprise Edition. Read only.

Boolean IsModified
Checks if component is modified since last save in Enterprise Edition. Read only.

Boolean IsOutdated
Checks if component is shared and there is a later version of the component on the Enterprise server in Enterprise Edition. Read only.

Boolean IsReadOnly
Checks if component is readonly (eg, shared and not checked out) in Enterprise Edition. Read only.

Boolean IsShared
Checks if component is shared on an Enterprise server in Enterprise Edition. Read only.

Boolean IsWritable
Checks if component is writable in Enterprise Edition. Read only.

Boolean Italic
Returns True if component font is italic and False otherwise, or modifies the font. Read/write.

LabelAlignMulti
Sets or returns alignment of multiline labels. Read/write.

LabelAlignX
Sets or returns horizontal label alignment. Read/write.

LabelAlignY
Sets or returns vertical label alignment. Read/write.

Format LabelBackground
Returns background color used for labels. Read only.

Format LabelBorder
Returns border color used for labels. Read only.

Boolean LabelEachBranch
Returns True if each label bound to a line outputs text near each branch of the line and False otherwise, or sets the output mode. Read/write.

Format LabelForeground
Returns foreground color used for labels. Read only.

Format LabelRotation
Returns rotation used for labels. Read only.

LabelSet LabelSet
Returns set of labels. Read only.

Format LabelSize
Returns size used for labels. Read only.

Format LabelStyle
Returns style used for labels. Read only.

Boolean LeftToRight
Returns True if each label bound to a line is automatically oriented left-to-right and False otherwise, or sets the orientation. Read/write.

Number LineCharacterSpacing
Sets or returns line character spacing, in points. Read/write.

Number LineInteriorSpacing
Sets or returns spacing between labels on the same line, in points. Read/write.

Number LineOffset
Sets or returns line offset, in points. Read/write.

Boolean MultipleLabelsPerBranch
Returns True if the system places multiple labels along a line and False otherwise, or modifies this behavior. Read/write.

String Name
Sets or returns component name. Read/write.

String Note
Sets or returns notes associated with the component. Read/write.

Boolean OptimizeLabelAlignX
Returns True if the system optimizes horizontal alignment for point labels and False otherwise, or allows or disallows optimization. Read/write.

Boolean OptimizeLabelAlignY
Returns True if the system optimizes vertical alignment for point labels and False otherwise, or allows or disallows optimization. Read/write.

ComponentSet OwnedComponentSet
Returns set of components bound to this component. Read only.

ComponentSet OwnedLayoutSet
Returns set of layouts bound to this component. Read only.

Component Owner
Returns owner component if any. Read only.

Any Parent
Returns parent object. Read only.
Boolean PerLabelFormat
Returns True if per-label format is enabled and False otherwise, or enables/disables per-label format. Read/write.

Number PointOffsetX
Sets or returns point offset by X, in points. Read/write.

Number PointOffsetY
Sets or returns point offset by Y, in points. Read/write.

Boolean ReduceShortLinesToPoints
Returns True if the system renders labels bound to lines that are too short to fit label text as though the labels were bound to the center of the first line branch and False otherwise, or modifies this behavior. Read / write.

Number RemoteVersion
Returns version of shared component available on the Enterprise server in Enterprise Edition. Read only.

Boolean ResolveOverlaps
Returns True if labels overlapping other labels are not displayed and False otherwise, or sets the overlap control mode. Read/write.

LabelSelectionSet SavedSelectionSet
Returns set of saved selections. Read only.

LabelSet Selection
Returns set of selected labels. Read only.

String Server
Returns server identification string for shared component in Enterprise Edition. Read only.

String ShareStatus
Returns textual report on component sharing status in Enterprise Edition. Read only.

Number Spacing
Sets or returns spacing between labels, in points. Read/write.

Boolean StrikeOut
Returns True if component font is striked out and False otherwise, or modifies the font. Read/write.

Boolean Synchronized
Returns True if the component automatically creates a new label for each new drawing object and False otherwise, or modifies this behavior, for a labels component bound to a drawing. Read/write.

Text Text
Sets or returns label text for label components bound to drawings. Read/write.

ComponentType Type
Returns component type. Read only.

String TypeName
Returns textual representation of the component type. Read only.

Boolean Underline
Returns True if component font is underlined and False otherwise, or modifies the font. Read/write.
Number Version
Returns version of shared component within the local project in Enterprise Edition. Read only.

ViewSet ViewSet
Returns set of views associated with this component. Read only.

Number ZoomMax
Sets or returns maximum zoom level. Read/write.

Number ZoomMin
Sets or returns minimum zoom level. Read/write.

Number ZoomRender
Sets or returns render zoom level. Read/write.

Methods

CheckIn()
Checks in changes made to a shared component to the Enterprise server in Enterprise Edition.

CheckOut()
Checks out a shared component for editing in Enterprise Edition.

Clear(Boolean selection)
Clears entire component or selection within the component. Calling the method without parameters clears entire component.

Copy(Boolean selection)
Copies entire component or selection within the component into the Clipboard. Calling the method without parameters copies entire component.

Cut(Boolean selection)
Copies entire component or selection within the component into the Clipboard and then clears it. Calling the method without parameters cuts entire component.

GetLatestVersion()
Gets the latest version of a shared component from the Enterprise server in Enterprise Edition.

Boolean isEmpty()
Returns True if component contains no labels and False otherwise.

Open()
Opens component.

OpenInNewWindow()
Opens component in new window.

Paste(Boolean replace)
Pastes the contents of the Clipboard into the component replacing current selection or leaving it intact. Calling the method without parameters replaces selection.

Print(Boolean useDialog)
Prints component.

ProjectTo(CoordinateSystem system)
Projects component to another coordinate system.

RenderAreaTo(String name, Number width, Number height, Rect area, Boolean clean)
Renders component area into an image with given name, width and height. If last parameter is True, legend, north arrow and scale bar are not rendered even if they are on.

RenderTo(String name, Number width, Number height, Boolean clean)
Render component into an image with given name, width and height. If last parameter is True, legend, north arrow and scale bar are not rendered even if they are turned on.

SelectAll()
Selects all labels.

SelectInverse()
Selects labels that were not selected and vice versa.

SelectNone()
Unselects all labels.

UndoCheckOut()
Undoes changes made to a shared component and gets the latest version of the component from the Enterprise server in Enterprise Edition.

Obtained From
Document (NewLabels)

See Also
Scripting Reference
LabelSelectionSet Object
Set of saves selections within labels.

Properties

Application Application
Returns application. Read only.

Number Count
Returns number of selections in set. Read only.

LabelSet Item(Number index)
Returns selection with given index. Read only.

Any Parent
Returns parent object. Read only.

Methods

Number ItemByMask(Number mask)
Returns index of selection with given mask value or -1.

Number ItemByName(String name)
Returns index of selection with given name or -1.

Obtained From

Labels (SavedSelectionSet)

See Also

Scripting Reference
LabelSet Object
Set of labels.

Properties

Application Application
Returns application. Read only.

Rect Box
Returns bounding box of all labels in set. Read only.

Number Count
Returns number of labels in set. Read only.

Label Item(Number index)
Returns label with given index. Read only.

Label LastAdded
Returns last added label if any. Read only.

Number Mask
Returns mask value. Read only.

String Name
Returns name of set. Read only.

Component OwnerComponent
Returns labels component containing this set. Read only.

Any Parent
Returns parent object. Read only.

Methods

Add(String text, Branch branch)
Add(String text, BranchSet branchSet)
Add(String text, Geom geom)
Add(String text, Point point)
Add(String text, PointSet pointSet)
Appends label to label set of stand-alone labels component.

Add(Object object, Branch branch)
Add(Object object, BranchSet branchSet)
Add(Object object, Geom geom)
Add(Object object, Point point)
Add(Object object, PointSet pointSet)
Appends label to label set of labels component bound to drawing.

Boolean IsSystem()
Returns True if label set is system and False otherwise.

Number ItemByID(Number id)
Returns index of label with given ID or -1.

Remove(Number index)
Removes label with given index.

RemoveAll()
Removes all labels.

Obtained From

Labels (LabelSet, Selection)
LabelSelectionSet (Item)

See Also

Scripting Reference
LabelWindow Object
Labels window.

Properties

Component ActiveComponent
Returns labels component shown in window. Read only.

Label ActiveLabel
Returns label selected for editing. Read only.

Application Application
Returns application. Read only.

Rect Bounds
Returns the area covered by the window, in the coordinate system of the displayed component. Read only.

Component Component
Returns labels component shown in window. Read only.

Number Height
Returns window height in pixels. Read only.

Point Location
Sets or returns viewport center. Read/write.

Point LocationLatLon
Sets or returns viewport center in lat/lon coordinates. Read/write.

Any Parent
Returns parent object. Read only.

Number Scale
Sets or returns viewport scale (projected components only). Read/write.

Number ScaleInternal
Sets or returns internal pixel-to-data ratio. Read/write.

Number Width
Returns window width in pixels. Read only.

Methods

Close()
Closes window.

Boolean HasActiveLabel()
Returns True if some label within the window is selected for editing and False otherwise.

Boolean HasLocation()
Returns True if viewport center can be modified and False otherwise.
Boolean HasScale()
Returns True if viewport scale can be obtained or modified and False otherwise.

Boolean HasScaleInternal()
Returns True.

MoveTo(Branch branch)
MoveTo(BranchSet branchSet)
MoveTo(Component component)
MoveTo(Geom geom)
MoveTo(Label label)
MoveTo(LabelSet labelSet)
MoveTo(Object object)
MoveTo(ObjectSet objectSet)
MoveTo(Pixel pixel)
MoveTo(PixelSet pixelSet)
MoveTo(Point point)
MoveTo(PointSet pointSet)
MoveTo(Rect rect)
Centers viewport on given object or set of objects. Branches, geoms, points, rects and their sets are assumed to be in the coordinate system of the displayed component.

MoveToLocation(Point point, Boolean latLon)
Centers viewport on given location. LatLon parameter is optional.

Refresh()
Updates window contents.

RenderTo(String name)
Renders window into an image with given name.

ZoomTo(Branch branch)
ZoomTo(BranchSet branchSet)
ZoomTo(Component component)
ZoomTo(Geom geom)
ZoomTo(Label label)
ZoomTo(LabelSet labelSet)
ZoomTo(Object object)
ZoomTo(ObjectSet objectSet)
ZoomTo(Pixel pixel)
ZoomTo(PixelSet pixelSet)
ZoomTo(Point point)
ZoomTo(PointSet pointSet)
ZoomTo(Rect rect)
Zooms to given object or set of objects. Branches, geoms, points, rects and their sets are assumed to be in the coordinate system of the displayed component. If the bounding box of the object is a point (for example, if the object is a point), centers viewport on given object.

 Obtained From

WindowSet (Item, ActiveWindow)

See Also
Scripting Reference
Layer Object
Map layer.

Properties

Application Application
Returns application. Read only.

Boolean Clickable
Sets or returns the "clickable" restriction of a layer. Read/write.

Component Component
Returns reference component. Read only.

Boolean Editable
Sets or returns the "editable" restriction of a layer. Read/write.

Number Order
Sets or returns layer position in the display stack, from 0 (the topmost layer) to the number of layers minus 1 (the bottommost layer). Read/write.

Number Opacity
Sets or returns layer opacity. Opacity is a value from 100 (opaque) to 0 (transparent). Read/write.

Component OwnerComponent
Returns map component containing this layer if any. Read only.

Any Parent
Returns parent object. Read only.

Boolean Selectable
Sets or returns the "selectable" restriction of a layer. Read/write.

Boolean Snappable
Sets or returns the "snappable" restriction of a layer. Read/write.

Boolean Visible
Sets or returns layer visibility state. Read/write.

Obtained From

Document (NewLayer)
LayerSet (Item, LastAdded)

See Also

Scripting Reference
LayerSet Object
Set of map layers.

Properties

Application Application
Returns application. Read only.

Number Count
Returns number of layers in set. Read only.

Layer Item(Number index)
Layer Item(String name)
Returns layer with given index or name. Read only.

Layer LastAdded
Returns last added layer if any. Read only.

Component OwnerComponent
Returns map component containing this set of layers if any. Read only.

Any Parent
Returns parent object. Read only.

Methods

Add(Layer layer, Number position)
Appends layer to layer set at given position. If the position argument is omitted, the layer is appended to the top of the display stack.

Number ItemByID(Number id)
Returns index of layer with given ID or -1.

Number ItemByName(String name)
Returns index of layer with given name or -1.

Remove(Number index)
Removes layer with given index.

Obtained From

Map (LayerSet)

See Also

Scripting Reference
**Layout Object**
Layout component.

**Properties**

*Application Application*
Returns application. Read only.

*Color BackgroundColor*
Sets or returns background color. Read / write.

*Boolean Cached*
Sets or returns caching option for shared component in Enterprise Edition. Read/write.

*String Description*
Sets or returns component description. Read/write.

*String DeviceName*
Printer name. Read only.

*String DevicePort*
Printer port name. Read only.

*Document Document*
Returns owner document. Read only.

*LayoutEntrySet EntrySet*
Set of layout entries. Read only.

*Folder Folder*
Sets or returns containing folder. Read/write.

*Number ID*
Returns unique component ID. Read only.

*Boolean IsCheckedOut*
Checks if component is shared and checked out in Enterprise Edition. Read only.

*Boolean IsCheckedOutRemotely*
Checks if component is shared and checked out by another user in Enterprise Edition. Read only.

*Boolean IsModified*
Checks if component is modified since last save in Enterprise Edition. Read only.

*Boolean IsOutdated*
Checks if component is shared and there is a later version of the component on the Enterprise server in Enterprise Edition. Read only.

*Boolean IsReadOnly*
Checks if component is readonly (eg, shared and not checked out) in Enterprise Edition. Read only.

*Boolean IsShared*
Checks if component is shared on an Enterprise server in Enterprise Edition. Read only.
Boolean IsWritable
Checks if component is writable in Enterprise Edition. Read only.

String Name
Sets or returns component name. Read/write.

String Note
Sets or returns notes associated with the component. Read/write.

ComponentSet OwnedComponentSet
Returns set of components bound to this component. Read only.

Component Owner
Returns owner component if any. Read only.

String PageFormat
Returns textual name of page format. Read only.

Number PageMarginBottom
Bottom print margin in points. Read only.

Number PageMarginLeft
Left print margin in points. Read only.

Number PageMarginRight
Right print margin in points. Read only.

Number PageMarginTop
Top page margin in points. Read only.

Number PageNonPrintableBottom
Bottom side of a non-printable area in points. Read only.

Number PageNonPrintableLeft
Left side of a non-printable area in points. Read only.

Number PageNonPrintableRight
Right side of a non-printable area in points. Read only.

Number PageNonPrintableTop
Top side of a non-printable area in points. Read only.

Number PageResolutionX
Horizontal page resolution in dots per inch (DPI). Read only.

Number PageResolutionY
Vertical page resolution in dots per inch (DPI). Read only.

Number PagesByX
Sets or returns number of pages by X axis. Read/write.

Number PagesByY
Sets or returns number of pages by Y axis. Read/write.

**Number PageSizeX**
Horizontal size of printable area less margins in points. Read only.

**Number PageSizeY**
Vertical size of printable area less margins in points. Read only.

**Any Parent**
Returns parent object. Read only.

**Number RemoteVersion**
Returns version of shared component available on the Enterprise server in Enterprise Edition. Read only.

**String Server**
Returns server identification string for shared component in Enterprise Edition. Read only.

**String ShareStatus**
Returns textual report on component sharing status in Enterprise Edition. Read only.

**ComponentType Type**
Returns component type. Read only.

**String TypeName**
Returns textual representation of the component type. Read only.

**Number Version**
Returns version of shared component within the local project in Enterprise Edition. Read only.

**Methods**

**CheckIn()**
Checks in changes made to a shared component to the Enterprise server in Enterprise Edition.

**CheckOut()**
Checks out a shared component for editing in Enterprise Edition.

**Clear(Boolean selection)**
Clears entire component or selection within the component. Calling the method without parameters clears entire component.

**Copy(Boolean selection)**
Copies entire component or selection within the component into the Clipboard. Calling the method without parameters copies entire component.

**Cut(Boolean selection)**
Copies entire component or selection within the component into the Clipboard and then clears it. Calling the method without parameters cuts entire component.

**GetLatestVersion()**
Gets the latest version of a shared component from the Enterprise server in Enterprise Edition.

**Boolean IsEmpty()**
Returns True if layout does not contain any printable objects and False otherwise.
Load(String template)
   Loads layout template with given name.

LoadFrom(String xml)
   Loads layout template from given XML string.

LoadFromFile(String path)
   Loads layout template from given XML file.

Open()
   Opens component.

OpenInNewWindow()
   Opens component in new window.

Paste(Boolean replace)
   Pastes the contents of the Clipboard into the component replacing current selection or leaving it intact. Calling
   the method without parameters replaces selection.

Print(Boolean useDialog)
   Prints component.

RenderTo(String name, Number dpi, Number page)
   Render specified page of a layout with given resolution into an image with given name. If page is omitted, the
   first page is rendered.

SaveToFile(String path)
   Saves layout template to given XML file.

String ToXML()
   Saves layout template to XML string.

UndoCheckOut()
   Undoes changes made to a shared component and gets the latest version of the component from the Enterprise
   server in Enterprise Edition.

Obtained From

Document (NewLayout)

See Also

Scripting Reference
LayoutEntry Object
Layout entry.

Properties

Application Application
Returns application. Read only.

Boolean AutoSelectColumns
Toggles between automatic and custom column selection mode for tables. Read/write.

LayoutState Background
Sets or returns background rendering option, can be set to LayoutStateCustom. Read/write.

Color BackgroundColor
Sets or returns background color for non-component entries. Read/write.

Color BackgroundCustomColor
Sets or returns background color for component entries. Used when Background property is set to LayoutStateCustom. Read/write.

LayoutBorder Border
Sets or returns border. Read/write.

Color BorderColor
Sets or returns border color. Read/write.

Boolean BorderDegMinSec
Sets or returns the use of a degree / minute / second format for border coordinates. Read/write.

Boolean BorderEachPage
Toggles between the use of a shared coordinate border or individual coordinate borders on each page. Read/write.

Font BorderFont
Sets or returns border font. Read/write.

Number BorderMargin
Sets or returns border margin in points. Read/write.

Boolean BorderOverlaps
Allows or disallows overlaps for border labels. Read/write.

Number BorderRounding
Sets or returns border rounding option. Read/write.

Number BorderStep
Sets or returns border interval. Read/write.

Unit BorderUnit
Sets or returns units used for border interval. Read/write.

Color Color
Sets or returns foreground color for non-component entries. Read/write.

Boolean ColumnHeaders
Toggles the use of column headers for table components. Read/write.

ColumnSet ColumnSet
Returns set of printed columns for a table component. Used when AutoSelectColumns property is set to False. Read only.

Component Component
Returns associated component. Read only.

LayoutState ControlPoints
Sets or returns control points rendering option for component entries. Read/write.

Font Font
Sets or returns font. Read/write.

LayoutState Graticule
Sets or returns graticule rendering option for component entries. Read/write.

LayoutState Grid
Sets or returns grid rendering option for component entries. Read/write.

Boolean GridLines
Toggles the use of grid lines for table components. Read/write.

Number ID
Returns unique entry ID. Read only.

LayoutState Legend
Sets or returns legend rendering option for component entries. Read/write.

Number Mask
Sets or returns entry mask. Read/write.

LayoutState NorthArrow
Sets or returns north arrow rendering option for component entries. Read/write.

LayoutEntry Owner
Returns owner entry. Used for legend, north arrow, scale bar and text entries. Read only.

Component OwnerComponent
Returns layout component containing this entry. Read only.

LayoutPaging Paging
Sets or returns layout paging mode. Read/write.

String PagingFilter
Sets or returns layout paging filter (eg, "1-5,8"). Read/write.

Any Parent
Returns parent object. Read only.
LayoutState ScaleBar
Sets or returns scale bar rendering option for component entries. Read/write.

LayoutScope Scope
Sets or returns scope for component entries. Read/write.

Rect ScopeArea
Sets or returns area covered by the entry. Setting an area for entries whose scope is different from LayoutScopeBox or LayoutScopeScale fails. Read/write.

Point ScopeCenter
Sets or returns center location of the entry. Setting a location for entries whose scope is different from LayoutScopeBox or LayoutScopeScale fails. Read/write.

String ScopeDetail
Sets or returns name of layer, saved selection or view for component entries. Read/write.

Number ScopeScale
Sets or returns absolute scale of the entry (10 means a scale of 1:10 and 0.5 means a scale of 2:1). Setting a scale for entries whose scope is different from LayoutScopeBox or LayoutScopeScale fails. Setting or getting a scale for entries whose components are in lat/lon fails. Read/write.

Number ScopeScaleInternal
Sets or returns internal scale of the entry in native coordinate system units per inch. Setting a scale for entries whose scope is different from LayoutScopeBox or LayoutScopeScale fails. Read/write.

Boolean Selected
Selects or deselects entry or checks if entry is selected. Read/write.

Boolean ShadeAlternateRows
Toggles the shading of alternate rows for table components. Read/write.

Number SizeX
Returns horizontal size of entry in points. Read only.

Number SizeY
Returns vertical size of entry in points. Read only.

String Text
Sets or returns text of a text entry. Read/write.

LabelAlignMulti TextAlignMulti
Sets or returns multiline alignment of a text entry. Read/write.

LabelAlignX TextAlignX
Sets or returns horizontal alignment of a text entry. Read/write.

LabelAlignY TextAlignY
Sets or returns vertical alignment of a text entry. Read/write.

Number TextDecimals
Sets or returns number of decimal points to use for a text entry. Read/write.
LayoutType

Returns entry type. Read only.

Number X

Returns left entry coordinate in points. Read only.

Number Y

Returns top entry coordinate in points. Read only.

Number ZOrder

Returns entry Z order. Read only.

**Methods**

Boolean HasOwner

Returns True if entry has an owner and False otherwise.

Boolean IsMasked(Number mask)

Returns True if entry has given mask or masks and False otherwise.

Boolean isSelected()

Returns True if entry is selected and False otherwise. This method is deprecated. Use the Selected property instead.

Boolean IsVisible()

Returns True if entry is visible and False otherwise.

MoveTo(Number x, Number y, Number sizeX, Number sizeY)

Moves entry to specified rectangle. All parameters are in points. Both X and Y size should be greater than zero.

**Obtained From**

- LayoutEntry (Owner)
- LayoutEntrySet (Item)
- LayoutWindow (ActiveEntry)

**See Also**

Scripting Reference
**LayoutEntrySet Object**
Set of layout entries.

**Properties**

- Application: Application
  Returns application. Read only.

- Number: Count
  Returns number of entries in set. Read only.

- LayoutEntry: Item(Number index)
  Returns entry with specified index. Read only.

- Component: OwnerComponent
  Returns layout component containing this set. Read only.

- Any: Parent
  Returns parent object. Read only.

**Methods**

- Add(LayoutType type)

- Add(LayoutType type, Number ownerID)

- Add(LayoutType type, LayoutEntry owner)
  Adds new entry with given type. Last two forms bound entry to a given owner entry.

- AddComponent(Component component)
  Adds new entry for given component.

- Number: ItemByID(Number id)
  Returns index of entry with given ID or -1.

- Remove(Number index)
  Removes entry with given index.

- RemoveAll()
  Removes all layout entries.

**Obtained From**

- Layout (EntrySet)

**See Also**

Scripting Reference
LayoutWindow Object
Layout window.

Properties

Component ActiveComponent
  Returns layout component shown in window. Read only.

LayoutEntry ActiveEntry
  Returns layout entry selected for editing. Read only.

Application Application
  Returns application. Read only.

Rect Bounds
  Returns the area covered by the window. Read only.

Component Component
  Returns layout component shown in window. Read only.

Number Height
  Returns window height in pixels. Read only.

Any Parent
  Returns parent object. Read only.

Number Scale
  Sets or returns display scale. Read/write.

Number ScaleInternal
  Sets or returns display scale in pixels per point (1/72 inch). Read/write.

Number Width
  Returns window width in pixels. Read only.

Methods

Close()
  Closes window.

Boolean HasActiveEntry()
  Returns True if some entry within the window is selected for editing and False otherwise.

Boolean HasScale()
  Returns True since layout window always has some scale.

Boolean HasScaleInternal()
  Returns True.

MoveTo(Branch branch)
MoveTo(BranchSet branchSet)
MoveTo(Geom geom)
MoveTo(LayoutEntry entry)
MoveTo(LayoutEntrySet entrySet)
MoveTo(Point point)
MoveTo(PointSet pointSet)
MoveTo(Rect rect)

Centers viewport on given object or set of objects. Branches, geoms, points, rects and their sets are assumed to be in the coordinate system of the page.

Refresh()

Updates window contents.

RenderTo(String name)

Renders window into an image with given name.

ZoomTo(Branch branch)
ZoomTo(BranchSet branchSet)
ZoomTo(Geom geom)
ZoomTo(LayoutEntry entry)
ZoomTo(LayoutEntrySet entrySet)
ZoomTo(Point point)
ZoomTo(PointSet pointSet)
ZoomTo(Rect rect)

Zooms to given object or set of objects. Branches, geoms, points, rects and their sets are assumed to be in the coordinate system of the page. If the bounding box of the object is a point (for example, if the object is a point), centers viewport on given object.

Obtained From

WindowSet (Item, ActiveWindow)

See Also

Scripting Reference
LookupValue Object
Value of the lookup column.

Properties

Application
Application
Returns application. Read only.

String Caption
Sets or returns value caption. Read/write.

Any Parent
Returns parent object. Read only.

Variant Value
Sets or returns value. Read/write.

Obtained From

ColumnLookupSet (Item, LastAdded, NewLookupValue)

See Also

Scripting Reference
**Map Object**
Map component.

**Properties**

Application Application
Returns application. Read only.

Color BackgroundColor
Sets or returns background color. Read / write.

Rect Box
Returns bounding box of all labels, objects and pixels in all layers. Read only.

Boolean Cached
Sets or returns caching option for shared component in Enterprise Edition. Read/write.

ControlPointSet ControlPointSet
Returns set of control points. Read only.

CoordinateSystem CoordinateSystem
Sets or returns coordinate system. Read/write.

View DefaultView
Returns default view. Read only.

String Description
Sets or returns component description. Read/write.

Document Document
Returns owner document. Read only.

Folder Folder
Sets or returns containing folder. Read/write.

Number ID
Returns unique component ID. Read only.

Boolean IsCheckedOut
Checks if component is shared and checked out in Enterprise Edition. Read only.

Boolean IsCheckedOutRemotely
Checks if component is shared and checked out by another user in Enterprise Edition. Read only.

Boolean IsModified
Checks if component is modified since last save in Enterprise Edition. Read only.

Boolean IsOutdated
Checks if component is shared and there is a later version of the component on the Enterprise server in Enterprise Edition. Read only.

Boolean IsReadOnly
Checks if component is readonly (eg, shared and not checked out) in Enterprise Edition. Read only.
Boolean IsShared
Checks if component is shared on an Enterprise server in Enterprise Edition. Read only.

Boolean IsWritable
Checks if component is writable in Enterprise Edition. Read only.

LayerSet LayerSet
Returns set of map layers. Read only.

String Name
Sets or returns component name. Read/write.

String Note
Sets or returns notes associated with the component. Read/write.

ComponentSet OwnedComponentSet
Returns set of components bound to this component. Read only.

ComponentSet OwnedLayoutSet
Returns set of layouts bound to this component. Read only.

Component Owner
Returns owner component if any. Read only.

Any Parent
Returns parent object. Read only.

Number RemoteVersion
Returns version of shared component available on the Enterprise server in Enterprise Edition. Read only.

MapSelectionSet SavedSelectionSet
Returns set of saved selections. Read only.

ObjectSet Selection
Returns set of selected drawing objects. Read only.

String Server
Returns server identification string for shared component in Enterprise Edition. Read only.

String ShareStatus
Returns textual report on component sharing status in Enterprise Edition. Read only.

ComponentType Type
Returns component type. Read only.

String TypeName
Returns textual representation of the component type. Read only.

Number Version
Returns version of shared component within the local project in Enterprise Edition. Read only.

ViewSet ViewSet
Returns set of views associated with this component. Read only.
Methods

CheckIn()
Checks in changes made to a shared component to the Enterprise server in Enterprise Edition.

CheckOut()
Checks out a shared component for editing in Enterprise Edition.

Copy()
Copies component into the Clipboard.

GetLatestVersion()
Gets the latest version of a shared component from the Enterprise server in Enterprise Edition.

Boolean IsEmpty()
Returns True if map contains no layers and False otherwise.

Open()
Opens component.

OpenInNewWindow()
Opens component in new window.

Print(Boolean useDialog)
Prints component.

RenderAreaTo(String name, Number width, Number height, Rect area, Boolean clean)
Renders component area into an image with given name, width and height. If last parameter is True, legend, north arrow and scale bar are not rendered even if they are on.

RenderTo(String name, Number width, Number height, Boolean clean)
Render component into an image with given name, width and height. If last parameter is True, legend, north arrow and scale bar are not rendered even if they are turned on.

UndoCheckOut()
Undoes changes made to a shared component and gets the latest version of the component from the Enterprise server in Enterprise Edition.

Obtained From

Document (NewMap)

See Also

Scripting Reference
MapSelectionSet Object
Set of saved selections within map.

Properties

Application Application
  Returns application. Read only.

Number Count
  Returns number of selections in set. Read only.

ObjectSet Item(Number index)
  Returns selection with given index. Read only.

Any Parent
  Returns parent object. Read only.

Methods

Number ItemByMask(Number mask)
  Returns index of selection with given mask value or -1.

Number ItemByName(String name)
  Returns index of selection with given name or -1.

Obtained From

Map (SavedSelectionSet)

See Also

Scripting Reference
MapServer Object
Mapserver object (Internet Map Server).

Properties

Application Application
Returns application. Read only.

Component Component
Returns served component. Read only.

String Copyright
Sets or returns copyright text. Read/write.

Number CurrentSessions
Returns the number of currently connected users. Can not be less than 1 (for obvious reasons). Read only.

Number CX
Sets or returns width of visible area in pixels. Read/write.

Number CY
Sets or returns height of visible area in pixels. Read/write.

Rect DataBox
Returns the bounding box of all data in served component. Read only.

Document Document
Returns served document. Read only.

String Fields
Returns list of searchable fields separated by newline characters. Read only.

Date FirstHitDate
Returns the date the first MapServer object was created. Uses the local time of the server. Read only.

Number Hits
Returns the total number of created MapServer objects since startup. Read only.

Date LastHitDate
Returns the date the last MapServer object was created. Uses the local time of the server. Read only.

Date LastReloadDate
Returns the date of the last reload. Uses the local time of the server. Read only.

String Layers
Returns list of layers separated by newline characters. Read only.

Boolean LayerShown(Number layer)
Boolean LayerShown(String layer)
Returns True if layer with given index or name is shown and False otherwise. Read only.

String Location
Returns textual representation of current location. Read only.

String Queries
Returns list of queries separated by newline characters. Read only.

String QueryParameters(Number query)
String QueryParameters(String query)
Returns list of parameters for query with given index or name separated by newline characters. Read only.

MapServerRenderFormat RenderFormat
Sets or returns rendering format. Read/write.

String RenderOptions
Sets or returns rendering options, specified as a sequence of tokens separated by whitespace characters. Support for rendering options varies between different rendering formats. For the complete list of supported rendering options, see the IMS Config.txt Options topic. Read/write.

String RenderParameters(String name)
Sets or returns the parameters for the drawing with the given name linked from a parameter query. Read/write.

Number RenderQuality
Sets or returns rendering quality, in the range of 1 to 100, with 100 meaning maximum quality. Support for rendering quality varies between different rendering formats. Read/write.

String Result
Returns result of information request, query or search. Read only.

String ResultRaw
Returns result of information request, query or search as <table>...</table> HTML fragment. Read only.

String ResultXml
Returns result of information request, query or search in XML format. Read only.

String Scale
Returns textual representation of current scale. Read only.

String State
Returns state string. Read only.

String Subtitle
Sets or returns subtitle text. Read/write.

String Target
Returns redirection URL if any. Read only.

String Title
Sets or returns title text. Read/write.

Number ViewCenterX
Sets or returns the X coordinate of the center location in native coordinates of the served component. Read/write.

Number ViewCenterY
Sets or returns the Y coordinate of the center location in native coordinates of the served component. Read/write.

String Views
Returns list of views separated by newline characters. Read only.

Number ViewScaleX
Sets or returns the number of native data units per pixel in X direction. Read/write.

Number ViewScaleY
Sets or returns the number of native data units per pixel in Y direction. Read/write.

Methods

Boolean _GetLayerShown(Number layer)
Boolean _GetLayerShown(String layer)
Returns True if layer with given index or name is shown and False otherwise. Provided as a way to use the LayerShown property in languages that do not support parameterized properties.

String _GetQueryParameters(Number query)
String _GetQueryParameters(String query)
Returns list of parameters for query with given index or name separated by newline characters. Provided as a way to use the QueryParameters property in languages that do not support parameterized properties.

String _GetRenderParameters(String name)
Returns the parameters for the drawing with the given name linked from a parameter query. Provided as a way to use the RenderParameters property in languages that do not support parameterized properties.

_putRenderParameters(String name, String value)
Sets the parameters for the drawing with the given name linked from a parameter query. Provided as a way to use the RenderParameters property in languages that do not support parameterized properties.

Center(Number x, Number y)
Center view at given location.

Click(Number x, Number y)
Request information at given location.

Create(String config, String state, Object server)
Initialize web server with given configuration file, state and IIS server object. The server object is used to format HTML data and is optional.

CreateWithFile(String project, String component, String state, Object server, Boolean useCache)
Initialize web server with given MAP file, component name, state and IIS server object. The server object is used to format HTML data and is optional. The last parameter controls the use of an option cache. The option cache associates each set of options with the ID of the served component and other data, which does not usually need to be recomputed from scratch for each instance of the MapServer object. It might be desired to avoid using the option cache if a component served by the MapServer object is generated on the fly. Passing true or omitting the parameter uses the option cache. Passing false does not use the option cache.

CreateWithOpts(String configOpts, String state, Object server, Boolean useCache)
Initialize web server with given configuration options, state and IIS server object. The server object is used to format HTML data and is optional. The last parameter controls the use of an option cache (see the description of the CreateWithFile method for details).

MapServerOgcWms CreateOgcWmsDriver()
Creates new OGC WMS driver object linked to this MapServer object.
GoToView(String view)
Jump to view with given name.

Boolean HasInfoAt(Number x, Number y)
Returns True if clicking given location with the Info tool will yield some data and False otherwise.

Boolean HasLinkAt(Number x, Number y)
Returns True if clicking given location will follow a hyperlink and False otherwise.

Locate(String what, String where, Boolean zoom)
Search for given text in given field and zoom to found objects if any. The last parameter is optional.

Number MapX(Number x)
Maps X coordinate of rendered image to that of served component.

Number MapY(Number y)
Maps Y coordinate of rendered image to that of served component.

Query(Number query, String parameters, Boolean force, Boolean zoom)
Perform query with given index or name using given set of parameters forcing recomputation and zooming the mapserver to the results of the query if desired. The values of the parameters are separated by newline characters. If the force parameter is omitted, the recomputation is not forced. If the zoom parameter is omitted, the mapserver zooms to the results of the query.

Reload()
Reloads all MAP files used by all web sites on a system the MapServer object has been instantiated on. Renders all MapServer objects in the current request queue inoperable.

Object Render()
Render current view to PNG file.

Object RenderLegend()
Render legend to PNG file.

Strafe(String direction)
Strafe view in given direction. The constants are "n", "ne", "e", "se", "s", "sw", "w" and "nw". Case does not matter

TurnLayer(Number layer, Boolean on)
Turn layer with given index or name on or off.

TurnLayer(String layer, Boolean on)

ZoomIn(Number x, Number y, Number factor)
Center current view at given location and increase zoom using given magnification factor. The magnification factor is a value from 1.1 to 4.0. If the magnification factor is omitted, the default magnification factor is used (normally, 2.0).

ZoomOut(Number x, Number y, Number factor)
Center current view at given location and decrease zoom using given magnification factor. The magnification factor is a value from 1.1 to 4.0. If the magnification factor is omitted, the default magnification factor is used (normally, 2.0).

ZoomToFit()
Zoom to see all data.
The RenderParameters property

The RenderParameters property is used to supply parameters to drawings linked from parameter queries to facilitate the use of such drawings by multiple users. The passed parameters are not persisted in the state string and are only valid for the lifetime of a given instance of the MapServer object.

The MapServer object refreshes images and surfaces linked from parameter queries prior to rendering, using the values of the RenderParameters property.

The logic used by the MapServer object to refresh components linked from parameter queries will skip re-computing queries the parameters of which did not change. This increases performance for simple scenarios of using components linked from parameter queries, but requires writing additional refresh code for complex scenarios.

Example

Suppose we have an IMS site that displays a map containing a drawing of cities named Cities. The drawing is linked to a parameter query that takes XY coordinates entered by the user and returns all cities close to that XY location.

Adding the following lines prior to a call to the Render method of the MapServer object will make the system customize the display using the values provided by the user:

```javascript
var locX = parameter("locX", ""); // value provided by the user
var locY = parameter("locY", ""); // value provided by the user
if (locX != "" && locY != "")
    mapserver.RenderParameters("Cities") = locX + "\n" + locY;
```

Comments

Regarding the state property of the MapServer object, more precisely the viewport parameter: The viewport is composed of four separate variables, ScaleX, ScaleY, CenterX and CenterY. The ScaleX and ScaleY parameters store the number of native coordinate system units in a pixel in the X and Y directions respectively.

Let's say we have a map in some coordinate system. Let's say the coordinate system unit is in feet and the local scale parameters are 5 and 2. Therefore, one native coordinate system unit is 5 feet "wide" and 2 feet "tall". Let's say that both the ScaleX and ScaleY parameters of the viewport are 10. Therefore, one pixel in the IMS view covers an area of 50 feet x 20 feet.

Creating an instance of the MapServer object with an empty state string (for example, for the first time in a session for a particular user) automatically goes to the view named Startup (case insensitive) if such a view exists. See the Views pane topic.

See the IMS programming examples in the Free Stuff web page to understand how to use the MapServer object.

See Also

Scripting Reference
MapServerOgcWms Object
OGC WMS interface driver for Internet Map Server.

Properties

Application Application
  Returns application. Read only.

Any Parent
  Returns parent object. Read only.

Any Result
  Returns response data for the processed WMS command. Read only.

String ResultContentType
  Returns content type of the response data for the processed WMS command. Read only.

Methods

Boolean Handle(String command, String url)
  Process the WMS command. The second parameter is an URL of the ASP page (used to compose the server capabilities document) and is optional. Returns True if the command has been processed successfully and False otherwise.

Obtained From

MapServer (CreateOgcWmsDriver)

See Also

Scripting Reference
MapWindow Object
Map window.

Properties

Component ActiveComponent
Returns active layer component. Read only.

Label ActiveLabel
Returns label selected for editing. Read only.

Object ActiveObject
Returns object selected for editing. Read only.

Application Application
Returns application. Read only.

Rect Bounds
Returns the area covered by the window, in the coordinate system of the displayed component. Read only.

Component Component
Returns map component shown in window. Read only.

Number Height
Returns window height in pixels. Read only.

Point Location
Sets or returns viewport center. Read/write.

Point LocationLatLon
Sets or returns viewport center in lat/lon coordinates. Read/write.

Any Parent
Returns parent object. Read only.

Number Scale
Sets or returns viewport scale (projected components only). Read/write.

Number ScaleInternal
Sets or returns internal pixel-to-data ratio. Read/write.

Number Width
Returns window width in pixels. Read only.

Methods

Close()
Closes window.

Boolean HasActiveLabel()
Returns True if some label within the window is selected for editing and False otherwise.
Boolean HasActiveObject()
Returns True if some object within the window is selected for editing and False otherwise.

Boolean HasLocation()
Returns True if viewport center can be modified and False otherwise.

Boolean HasScale()
Returns True if viewport scale can be obtained or modified and False otherwise.

Boolean HasScaleInternal()
Returns True.

MoveTo(Branch branch)
MoveTo(BranchSet branchSet)
MoveTo(Component component)
MoveTo(Geom geom)
MoveTo(Label label)
MoveTo(LabelSet labelSet)
MoveTo(Object object)
MoveTo(ObjectSet objectSet)
MoveTo(Pixel pixel)
MoveTo(PixelSet pixelSet)
MoveTo(Point point)
MoveTo(PointSet pointSet)
MoveTo(Rect rect)
Centers viewport on given object or set of objects. Branches, geoms, points, rects and their sets are assumed to be in the coordinate system of the displayed component.

MoveToLocation(Point point, Boolean latLon)
Centers viewport on given location. LatLon parameter is optional.

Refresh()
Updates window contents.

RenderTo(String name)
Renders window into an image with given name.

ZoomTo(Branch branch)
ZoomTo(BranchSet branchSet)
ZoomTo(Component component)
ZoomTo(Geom geom)
ZoomTo(Label label)
ZoomTo(LabelSet labelSet)
ZoomTo(Object object)
ZoomTo(ObjectSet objectSet)
ZoomTo(Pixel pixel)
ZoomTo(PixelSet pixelSet)
ZoomTo(Point point)
ZoomTo(PointSet pointSet)
ZoomTo(Rect rect)
Zooms to given object or set of objects. Branches, geoms, points, rects and their sets are assumed to be in the coordinate system of the displayed component. If the bounding box of the object is a point (for example, if the object is a point), centers viewport on given object.
Obtained From

WindowSet (Item, ActiveWindow)

See Also

Scripting Reference
**Object**

Drawing object.

**Properties**

```
Application Application
Returns application. Read only.

Color BackColor
Returns background color. Read only.

Color BorderBackColor
Returns border background color. Read only.

Color BorderForeColor
Returns border foreground color. Read only.

Number BorderSize
Returns border size. Read only.

Number BorderStyle
Returns border style. Read only.

Geom ChangeGeomLocal
Gets the local version of the object metric in case the object belongs to a linked drawing and participates in an editing conflict. Read only.

Geom ChangeGeomRemote
Gets the remote version of the object metric in case the object belongs to a linked drawing and participates in an editing conflict. Read only.

Color ForeColor
Returns foreground color. Read only.

Geom Geom
Sets or returns object metric. Read/write.

Boolean HasLocalChanges
Returns True if the object belongs to a linked drawing and participates in an editing conflict, and False otherwise. Read only.

Number ID
Returns unique object ID. Read only.

Number Mask
Sets or returns object mask. Read/write.

Component OwnerComponent
Returns drawing component containing this object. Read only.

Any Parent
Returns parent object. Read only.
```
Record

Returns the record associated with the drawing object. Read only.

Boolean Selected

Selects or deselects object or checks if object is selected. Read/write.

Number Size

Returns object size. Read only.

Number Style

Returns object style. Read only.

ObjectType Type

Returns object type. Read only.

String TypeName

Returns textual representation of object type. Read only.

Methods

Boolean CommitLocalChanges()
Commits local changes to an object that belongs to a linked drawing and participates in an editing conflict, and uploads them to the drawing's data source. Returns True if, at the end of the method call, the object does not participate in an editing conflict and False otherwise.

Boolean DiscardLocalChanges()
Discards local changes to an object that belongs to a linked drawing and participates in an editing conflict, and retrieves the latest data for the object from the drawing's data source. Returns True if, at the end of the method call, the object does not participate in an editing conflict and False otherwise.

Boolean IsClean()
Returns True if object metric has been normalized and False otherwise.

Boolean IsConvex()
Returns True if area represented by the object is convex and False otherwise.

Boolean IsMasked(Number mask)
Returns True if object has given mask or masks and False otherwise.

Boolean IsSelected()
Returns True if object is selected and False otherwise. This method is deprecated. Use the Selected property instead.

Obtained From

DrawingWindow (ActiveObject)
EventArgs (Object)
Label (Link)
MapWindow (ActiveObject)
ObjectSet (Item, LastAdded)
Record (Object)

See Also
ObjectSet Object
Set of objects. Can represent all objects in a particular component or an arbitrary set of objects in one or more components. In the first case, the set is considered to contain objects, and the Add and Remove methods add and remove objects from the component. In the second case, the set is considered to contain references to objects, and the Add and Remove methods alter the content of the object set without adding or removing objects from components.

Properties
Application Application
Returns application. Read only.

Rect Box
Returns bounding box of all objects in set. Read only.

Number Count
Returns number of objects in set. Read only.

Number CountAreas
Returns number of areas in set. Read only.

Number CountLines
Returns number of lines in set. Read only.

Number CountPoints
Returns number of points in set. Read only.

GeomSet GeomSet
Returns geometric entities representing a set. Read only.

Object Item(Number index)
Returns object with given index. Read only.

Object LastAdded
Returns last added object if any. Read only.

Number Mask
Returns mask value. Read only.

String Name
Returns name of set. Read only.

Component OwnerComponent
Returns drawing component containing this set. Read only.

Any Parent
Returns parent object. Read only.

Methods
Add(Geom geom)
Creates new object with given metric and adds it to a set which contains objects.
Add(Object object)
Adds object to a set which contains references to objects.

Boolean IsStatic()
Returns True if object set is static and False otherwise.

Boolean IsSystem()
Returns True if object set is system and False otherwise.

Number ItemByID(Number id)
Returns index of object with given ID or -1.

Remove(Number index)
Removes object with given index.

RemoveAll()
Removes all objects.

Obtained From
Document (NewObjectSet)
Drawing (ObjectSet, Selection)
DrawingSelectionSet (Item)
Map (Selection)
MapSelectionSet (Item)

See Also

Scripting Reference
**Palette Object**
Palette component.

**Properties**

- **Application**
  Returns application. Read only.

- **Cached**
  Sets or returns caching option for shared component in Enterprise Edition. Read/write.

- **ColorSet**
  Returns set of palette colors. Read only.

- **Description**
  Sets or returns component description. Read/write.

- **Document**
  Returns owner document. Read only.

- **Folder**
  Sets or returns containing folder. Read/write.

- **ID**
  Returns unique component ID. Read only.

- **IsCheckedOut**
  Checks if component is shared and checked out in Enterprise Edition. Read only.

- **IsCheckedOutRemotely**
  Checks if component is shared and checked out by another user in Enterprise Edition. Read only.

- **IsModified**
  Checks if component is modified since last save in Enterprise Edition. Read only.

- **IsOutdated**
  Checks if component is shared and there is a later version of the component on the Enterprise server in Enterprise Edition. Read only.

- **IsReadOnly**
  Checks if component is readonly (e.g., shared and not checked out) in Enterprise Edition. Read only.

- **IsShared**
  Checks if component is shared on an Enterprise server in Enterprise Edition. Read only.

- **IsWritable**
  Checks if component is writable in Enterprise Edition. Read only.

- **Name**
  Sets or returns component name. Read/write.

- **Note**
  Sets or returns notes associated with the component. Read/write.
ComponentSet OwnedComponentSet
Returns set of components bound to this component. Read only.

ComponentSet OwnedLayoutSet
Returns set of layouts bound to this component. Read only.

Component Owner
Returns owner component if any. Read only.

Any Parent
Returns parent object. Read only.

Number RemoteVersion
Returns version of shared component available on the Enterprise server in Enterprise Edition. Read only.

PaletteSelectionSet SavedSelectionSet
Returns set of saved selections. Read only.

ColorSet Selection
Returns set of selected colors. Read only.

String Server
Returns server identification string for shared component in Enterprise Edition. Read only.

String ShareStatus
Returns textual report on component sharing status in Enterprise Edition. Read only.

ComponentType Type
Returns component type. Read only.

String TypeName
Returns textual representation of the component type. Read only.

Number Version
Returns version of shared component within the local project in Enterprise Edition. Read only.

**Methods**

CheckIn()
Checks in changes made to a shared component to the Enterprise server in Enterprise Edition.

CheckOut()
Checks out a shared component for editing in Enterprise Edition.

Clear(Boolean selection)
Clears entire component or selection within the component. Calling the method without parameters clears entire component.

Copy(Boolean selection)
Copies entire component or selection within the component into the Clipboard. Calling the method without parameters copies entire component.
Cut(Boolean selection)
Copies entire component or selection within the component into the Clipboard and then clears it. Calling the method without parameters cuts entire component.

GetLatestVersion()
Gets the latest version of a shared component from the Enterprise server in Enterprise Edition.

Open()
Opens component.

OpenInNewWindow()
Opens component in new window.

Paste(Boolean replace)
Pastes the contents of the Clipboard into the component replacing current selection or leaving it intact. Calling the method without parameters replaces selection.

Print(Boolean useDialog)
Prints component.

UndoCheckOut()
Undoes changes made to a shared component and gets the latest version of the component from the Enterprise server in Enterprise Edition.

**Obtained From**

Document (NewPalette)
Image (OwnedPalette)

**See Also**

Scripting Reference
PaletteSelectionSet Object
Set of saved selections within palette.

Properties

Application Application
Returns application. Read only.

Number Count
Returns number of selections in set. Read only.

ColorSet Item(Number index)
Returns selection with given index. Read only.

Any Parent
Returns parent object. Read only.

Methods

Number ItemByMask(Number mask)
Returns index of selection with given mask value or -1.

Number ItemByName(String name)
Returns index of selection with given name or -1.

Obtained From

Palette (SavedSelectionSet)

See Also

Scripting Reference
**Picture Object**

Picture.

**Properties**

Number Attributes
Returns picture attributes. Read only.

Number Height
Returns height of picture in pixels. Read only.

Number Type
Returns picture type. Read only.

Number Width
Returns width of picture in pixels. Read only.

**See Also**

Scripting Reference
**Pixel Object**
Pixel of image or surface.

**Properties**

Application Application
Returns application. Read only.

Number Channel(Number index)
Sets or returns value in given channel. Read/write.

Color Color
Sets or returns color. Read/write.

Number Mask
Sets or returns pixel mask. Read/write.

Component OwnerComponent
Returns image or surface component containing this pixel. Read only.

Any Parent
Returns parent object. Read only.

Boolean Selected
Selects or deselects pixel or checks if pixel is selected. Read/write.

Number Value
Sets or returns value associated with pixel. Read/write.

Variant ValueNative
Returns native value associated with pixel. Read only.

Boolean Visible
Sets or returns pixel visibility flag. Read/write.

Number X
Returns X coordinate of pixel within image or surface. Read only.

Number Y
Returns Y coordinate of pixel within image or surface. Read only.

**Methods**

Number _GetChannel(Number index)
Returns value in given channel. Provided as a way to use the Channel property in languages that do not support parameterized properties.

_PutChannel(Number index, Number value)
Sets value in given channel. Provided as a way to use the Channel property in languages that do not support parameterized properties.

Boolean IsInvisible()
Returns True if pixel is invisible and False otherwise. This method is deprecated. Use the Visible property instead.
Boolean IsMasked(Number mask)
Returns True if pixel has given mask or masks and False otherwise.

Boolean IsMissing()
Returns True if pixel is invisible and False otherwise. This method is deprecated. Use the Visible property instead.

Boolean IsSelected()
Returns True if pixel is selected and False otherwise. This method is deprecated. Use the Selected property instead.

Obtained From

PixelSet (Item)

See Also

Scripting Reference
PixelSet Object
Set of pixels.

Properties

Application Application
Returns application.  Read only.

Rect Box
Returns bounding box of all pixels.  Read only.

Number Count
Returns number of pixels in set.  Read only.

Pixel Item(Number index)
Returns pixel with given index.  Read only.

Number ItemByXY(Number x, Number y)
Returns index of pixel with given X and Y coordinates or -1.

Pixel LastAdded
    Returns last added pixel if any. Read only.

Number Mask
Returns mask value.  Read only.

String Name
Returns name of set.  Read only.

Component OwnerComponent
Returns image or surface component containing this set.  Read only.

Any Parent
Returns parent object.  Read only.

Number ValueMax
Returns maximum value of pixel in set.  Read only.

Number ValueMin
Returns minimum value of pixel in set.  Read only.

Methods

Number _GetItemByXY(Number x, Number y)
Returns index of pixel with given X and Y coordinates or -1. Provided as a way to use the ItemByXY property in languages that do not support parameterized properties.

Add(Number x, Number y)
Add a pixel at a given location to the selection or a saved selection.

Boolean IsSystem()
Returns True if pixel set is system and False otherwise.
Remove(Number index)
Removes a pixel with given index from the selection or a saved selection.

RemoveAll()
Clears the selection or a saved selection.

TransformWith(String formula)
Transforms pixel set according to specified formula. Only works on surfaces. Requires Surface Tools extension.

Obtained From
Image (PixelSet, Selection)
ImageSelectionSet (Item)
Surface (PixelSet, Selection)
SurfaceSelectionSet (Item)

See Also
Scripting Reference
Point Object
Location.

Properties

Application Application
Returns application. Read only.

Any Parent
Returns parent object. Read only.

Number X
Sets or returns X coordinate. Read/write.

Number Y
Sets or returns Y coordinate. Read/write.

Methods

Number BearingTo(Point point)
Returns bearing to another location.

Copy(Point point)
Copies given point.

Number DistanceTo(Point point)
Returns Euclidean distance to another location.

String ToTextUSNG(PointUSNGPrecision precision)
Outputs the coordinates of a lat/lon point in USNG format with given precision. If the precision parameter is omitted, uses 1 meter precision.

Obtained From

Application (NewPoint, NewPointFromUSNG)
Branch (Center)
BranchSet (Center)
ControlPoint (Location)
CoordinateSystem (ConvertForward, ConvertInverse)
DrawingWindow (Location, LocationLatLon)
EventArgs (LocationLatLon, LocationNative, LocationScreen)
Geocoder (LocateZip)
Geom (Center, CenterWeight)
GeomSet (Center)
Label (Center)
LabelSet (Center)
LabelWindow (Location, LocationLatLon)
LayoutEntry (ScopeCenter)
MapWindow (Location, LocationLatLon)
PointSet (Center, Item, LastAdd)
Rect (Center, FitPoint)
View (Center)
Window (GetLocation)
See Also

Scripting Reference
**PointSet Object**
Set of points.

**Properties**

Application Application
Returns application. Read only.

Rect Box
Sets or returns bounding box. Modifying this property scales and shifts the point set. Read/write.

Point Center
Sets or returns center of minimum enclosing circle. Modifying this property moves the point set. Read/write.

Number Count
Returns number of points in set. Read only.

Point Item(Number index)
Sets or returns point with given index. Read/write.

Point LastAdded
Returns last added point if any. Read only.

Any Parent
Returns parent object. Read only.

**Methods**

Add(Point point)
Appends point to set.

Copy(PointSet pointSet)
Copies given point set.

Remove(Number index)
Removes point with given index.

RemoveAll()
Removes all points.

**Obtained From**

Application (NewPointSet)
Branch (PointSet)
DSSAtom (Points)
Label (PointSet)

**See Also**

Scripting Reference
Profile Object
Profile component.

Properties

Application Application
  Returns application. Read only.

Number AvgHeight
  Returns average height of the profile. Read only.

Color BackgroundColor
  Sets or returns background color. Read / write.

Boolean Cached
  Sets or returns caching option for shared component in Enterprise Edition. Read/write.

ControlPointSet ControlPointSet
  Returns set of control points. Read only.

Boolean Coordinates
  Turns coordinate display on or off. Read/write.

CoordinateSystem CoordinateSystem
  Sets or returns coordinate system. Read/write.

Boolean DecimalDigits
  Sets or returns the number of decimal digits used for height reports. Read/write.

View DefaultView
  Returns default view. Read only.

String Description
  Sets or returns component description. Read/write.

Document Document
  Returns owner document. Read only.

Folder Folder
  Sets or returns containing folder. Read/write.

Number ID
  Returns unique component ID. Read only.

Boolean IsCheckedOut
  Checks if component is shared and checked out in Enterprise Edition. Read only.

Boolean IsCheckedOutRemotely
  Checks if component is shared and checked out by another user in Enterprise Edition. Read only.

Boolean IsModified
  Checks if component is modified since last save in Enterprise Edition. Read only.
Boolean IsOutdated
Checks if component is shared and there is a later version of the component on the Enterprise server in Enterprise Edition. Read only.

Boolean IsReadOnly
Checks if component is readonly (eg, shared and not checked out) in Enterprise Edition. Read only.

Boolean IsShared
Checks if component is shared on an Enterprise server in Enterprise Edition. Read only.

Boolean IsWritable
Checks if component is writable in Enterprise Edition. Read only.

Number MaxHeight
Returns maximum height within the profile. Read only.

Geom Metric
Sets or returns profile metric (can accept BranchSet, Branch, or PointSet when setting). Read/write.

Number MinHeight
Returns minimum height within the profile. Read only.

String Name
Sets or returns component name. Read/write.

String Note
Sets or returns notes associated with the component. Read/write.

ComponentSet OwnedComponentSet
Returns set of components bound to this component. Read only.

ComponentSet OwnedElevationSet
Returns set of elevations bound to this component. Read only.

ComponentSet OwnedLayoutSet
Returns set of layouts bound to this component. Read only.

Component Owner
Returns owner component if any. Read only.

Any Parent
Returns parent object. Read only.

Number RemoteVersion
Returns version of shared component available on the Enterprise server in Enterprise Edition. Read only.

String Server
Returns server identification string for shared component in Enterprise Edition. Read only.

String ShareStatus
Returns textual report on component sharing status in Enterprise Edition. Read only.
ComponentType Type
   Returns component type. Read only.

String TypeName
   Returns textual representation of the component type. Read only.

Number Version
   Returns version of shared component within the local project in Enterprise Edition. Read only.

ViewSet ViewSet
   Returns set of views associated with this component. Read only.

Number ZoomMax
   Sets or returns maximum zoom level. Read/write.

Number ZoomMin
   Sets or returns minimum zoom level. Read/write.

Number ZoomRender
   Sets or returns render zoom level. Read/write.

Methods

CheckIn()
   Checks in changes made to a shared component to the Enterprise server in Enterprise Edition.

CheckOut()
   Checks out a shared component for editing in Enterprise Edition.

Copy()
   Copies component into the Clipboard.

GetLatestVersion()
   Gets the latest version of a shared component from the Enterprise server in Enterprise Edition.

Boolean IsEmpty()
   Returns True if profile contains no entries and False otherwise.

Open()
   Opens component.

OpenInNewWindow()
   Opens component in new window.

Print(Boolean useDialog)
   Prints component.

RenderAreaTo(String name, Number width, Number height, Rect area, Boolean clean)
   Renders component area into an image with given name, width and height. If last parameter is True, legend,
   north arrow and scale bar are not rendered even if they are on.

RenderTo(String name, Number width, Number height, Boolean clean)
Render component into an image with given name, width and height. If last parameter is True, legend, north arrow and scale bar are not rendered even if they are turned on.

UndoCheckOut()
Undoes changes made to a shared component and gets the latest version of the component from the Enterprise server in Enterprise Edition.

Obtained From
Document (NewProfile)

See Also

Scripting Reference
**Progress Object**
Progress tracker.

**Properties**

*Application Application*
Returns application. Read only.

*String Caption*
Sets or returns caption text. Read/write.

*Number MaxPosition*
Sets or returns maximum progress value. Read/write.

*Any Parent*
Returns parent object. Read only.

*Number Position*
Sets or returns current progress value. Read/write.

*String Text*
Sets or returns tracking text. Read/write.

**Methods**

*Boolean Start()*
Starts tracking progress. Returns True if started successfully and False otherwise.

*Boolean Stop()*
Stops tracking progress. Returns True if progress was not running and False otherwise.

*Boolean Update()*
Updates progress window and watches for cancel events. Returns True if process has not been canceled and False otherwise.

**Obtained From**

*Application (NewProgress)*

**See Also**

Scripting Reference
Query Object
Query component.

Properties

Boolean Ansi
Sets or returns the ANSI-compatible syntax option. Read/write.

Application Application
Returns application. Read only.

Boolean Cached
Sets or returns caching option for shared component in Enterprise Edition. Read/write.

String Description
Sets or returns component description. Read/write.

Document Document
Returns owner document. Read only.

Folder Folder
Sets or returns containing folder. Read/write.

Number ID
Returns unique component ID. Read only.

Boolean IsCheckedOut
Checks if component is shared and checked out in Enterprise Edition. Read only.

Boolean IsCheckedOutRemotely
Checks if component is shared and checked out by another user in Enterprise Edition. Read only.

Boolean IsModified
Checks if component is modified since last save in Enterprise Edition. Read only.

Boolean IsOutdated
Checks if component is shared and there is a later version of the component on the Enterprise server in Enterprise Edition. Read only.

Boolean IsReadOnly
Checks if component is readonly (e.g., shared and not checked out) in Enterprise Edition. Read only.

Boolean IsShared
Checks if component is shared on an Enterprise server in Enterprise Edition. Read only.

Boolean IsWritable
Checks if component is writable in Enterprise Edition. Read only.

Number Length
Returns number of characters in query text. Read only.

String Name
Sets or returns component name. Read/write.
String Note
Sets or returns notes associated with the component. Read/write.

ComponentSet OwnedComponentSet
Returns set of components bound to this component. Read only.

ComponentSet OwnedLayoutSet
Returns set of layouts bound to this component. Read only.

Component Owner
Returns owner component if any. Read only.

QueryParameterSet ParameterSet
Returns set of query parameters. Read only.

Any Parent
Returns parent object. Read only.

QueryType QueryType
Returns type of query. Read only.

Number RemoteVersion
Returns version of shared component available on the Enterprise server in Enterprise Edition. Read only.

String Server
Returns server identification string for shared component in Enterprise Edition. Read only.

String ShareStatus
Returns textual report on component sharing status in Enterprise Edition. Read only.

Table Table
Returns query table. Read only.

String Text
Sets or returns query text. Read/write.

ComponentType Type
Returns component type. Read only.

String TypeName
Returns textual representation of the component type. Read only.

Number Version
Returns version of shared component within the local project in Enterprise Edition. Read only.

Methods

AddText(String text)
Appends text to the end of the component.

CheckIn()
Checks in changes made to a shared component to the Enterprise server in Enterprise Edition.
CheckOut()
Checks out a shared component for editing in Enterprise Edition.

Clear()
Clears component.

Copy()
Copies component into the Clipboard.

Cut()
Copies component into the Clipboard and then clears it.

Boolean EditParameters()
Edits parameters of a parameter query in a dialog. Returns True if editing has been accepted and False otherwise.

GetLatestVersion()
Gets the latest version of a shared component from the Enterprise server in Enterprise Edition.

Boolean IsEmpty(Boolean ignoreWhiteSpaces)
Returns True if component text is empty and False otherwise.

Boolean IsValid()
Returns True if query text does not contain syntax errors and False otherwise.

Open()
Opens component.

OpenInNewWindow()
Opens component in new window.

Paste(Boolean replace)
Pastes the contents of the Clipboard into the component replacing previous data or adding to it. Calling the method without parameters replaces data.

Print(Boolean useDialog)
Prints component.

Run()
Runs query.

RunEx(Boolean force)
Runs query forcing recomputation if desired.

UndoCheckOut()
Undoes changes made to a shared component and gets the latest version of the component from the Enterprise server in Enterprise Edition.

Obtained From
Document (NewQuery)

See Also
Scripting Reference
QueryParameter Object
Query parameter.

Properties

Application Application
Returns application. Read only.

String Name
Returns parameter name. Read only.

Any Parent
Returns parent object. Read only.

String Type
Returns parameter type. Read only.

String Value
Sets or returns parameter value. Read/write.

Obtained From

QueryParameterSet (Item)

See Also

Scripting Reference
QueryParameterSet Object
Set of query parameters.

Properties

Application Application
Returns application. Read only.

Number Count
Returns number of parameters in set. Read only.

QueryParameter Item(Number index)
QueryParameter Item(String name)
Returns parameter with given index or name. Read only.

Any Parent
Returns parent object. Read only.

Methods

Number ItemByName(String name)
Returns index of parameter with given name or -1.

Obtained From

Query (ParameterSet)

See Also

Scripting Reference
**RecentDataSource Object**
Connection information for a recently used data source.

**Properties**

**Application Application**
Returns application. Read only.

**ConnectionString**
Returns connection string. Read only.

**ConnectionType**
Returns connection type. For the list of supported connection types see the topic on the DataSource object. Read only.

**Name**
Returns data source name. Read only.

**Parent**
Returns parent object. Read only.

**Obtained From**

RecentDataSourceSet (Item)

**See Also**

Scripting Reference
**RecentDataSourceSet Object**
A set of recently used data sources.

**Properties**

- **Application Application**
  - Returns application. Read only.

- **Number Count**
  - Returns number of data sources in set. Read only.

- **RecentDataSource Item(Number index)**
- **RecentDataSource Item(String name)**
  - Returns data source with given index or name. Read only.

- **Any Parent**
  - Returns parent object. Read only.

**Methods**

- **Number ItemByName(String name)**
  - Returns the index of a data source with given name, or -1.

**Obtained From**

- Application (RecentDataSourceSet)

**See Also**

- Scripting Reference
**Record Object**
Table record.

**Properties**

- **Application**
  Returns application. Read only.

- **Variant Data(Column column)**
  **Variant Data(Number id)**
  **Variant Data(String name)**
  Sets or returns data for given column. Column can be supplied directly or by its ID or name. Read/write.

- **Variant DataByIndex(Number index)**
  Sets or returns data for column with given index. Read/write.

- **String DataText(Column column)**
  **String DataText(Number id)**
  **String DataText(String name)**
  Returns textual representation of data in given column. Column can be supplied directly or by its ID or name. Read only.

- **String DataTextByIndex(Number index)**
  Returns textual representation of data in column with given index. Read only.

- **Number ID**
  Returns unique record ID. Read only.

- **Number Index**
  Returns the index of this record in the record set. Read only.

- **Number Mask**
  Sets or returns record mask. Read/write.

- **Object**
  Returns the drawing object associated with the record. If the table is not bound to a drawing, returns an empty value. Read only.

- **Component OwnerComponent**
  Returns table component containing this record if any. Read only.

- **Any Parent**
  Returns parent object. Read only.

- **Boolean Selected**
  Selects or deselects record or checks if record is selected. Read/write.

**Methods**

- **Variant _GetData(Column column)**
- **Variant _GetData(Number id)**
- **Variant _GetData(String name)**
Programming Reference

Returns data for given column. Column can be supplied directly or by its ID or name. Provided as a way to use the Data property in languages that do not support parameterized properties.

Variant _GetDataByIndex(Number index)

Returns data for column with given index. Provided as a way to use the DataByIndex property in languages that do not support parameterized properties.

String _GetDataText(Column column)
String _GetDataText(Number id)
String _GetDataText(String name)

Returns textual representation of data in given column. Column can be supplied directly or by its ID or name. Provided as a way to use the DataText property in languages that do not support parameterized properties.

String _GetDataTextByIndex(Number index)

Returns textual representation of data in column with given index. Provided as a way to use the DataTextByIndex property in languages that do not support parameterized properties.

_PutData(Column column, Variant value)
_PutData(Number id, Variant value)
_PutData(String name, Variant value)

Sets data for given column. Column can be supplied directly or by its ID or name. Provided as a way to use the Data property in languages that do not support parameterized properties.

_PutDataByIndex(Number index, Variant value)

Sets data for column with given index. Provided as a way to use the DataByIndex property in languages that do not support parameterized properties.

Boolean IsMasked(Number mask)

Returns True if record has given mask or masks and False otherwise.

Boolean IsSelected()

Returns True if record is selected and False otherwise. This method is deprecated. Use the Selected property instead.

Obtained From

Object (Record)
RecordSet (Item, LastAdded)
TableWindow (ActiveRecord)

See Also

Scripting Reference
RecordSet Object
Set of table records.

Properties

Application Application
Returns application. Read only.

Number Count
Returns number of records in set. Read only.

Record Item(Number index)
Returns record with given index. Read only.

Record LastAdded
Returns last added record if any. Read only.

Component OwnerComponent
Returns table component containing this record if any. Read only.

Any Parent
Returns parent object. Read only.

Methods

Add(Record record)
Appends record to the record set. Adds a new record to the table if the record set represents all table records. Selects the record or adds the record to the saved selection if the record set represents the selection or a saved selection.

AddNew()
Appends new record to set.

Number Average(Column column)
Number Average(Number id)
Number Average(String name)
Returns average value of given column. Column can be supplied directly or by its ID or name.

RecordSet Bottom(Column column, Number count)
RecordSet Bottom(Number id, Number count)
RecordSet Bottom(String name, Number count)
Returns given number of bottom records for given column. Column can be supplied directly or by its ID or name.

Number CenterCorrelation(Column column, Column columnSecond)
Number CenterCorrelation(Number id, Number idSecond)
Number CenterCorrelation(String name, String stringSecond)
Returns central correlation of given pair of columns. Columns can be supplied directly or by their IDs or names.

Number CenterCovariation(Column column, Column columnSecond)
Number CenterCovariation(Number id, Number idSecond)
Number CenterCovariation(String name, String stringSecond)
Returns central covariance of given pair of columns. Columns can be supplied directly or by their IDs or names.
Number CenterMoment(Column column, Number order)
Number CenterMoment(Number id, Number order)
Number CenterMoment(String name, Number order)
Returns central momentum of given order for given column. Column can be supplied directly or by its ID or name.

RecordSet Containing(Column column, Variant value)
RecordSet Containing(Number id, Variant value)
RecordSet Containing(String name, Variant value)
Returns records containing given value in given column. Column can be supplied directly or by its ID or name.

RecordSet ContainingMatch(Column column, String pattern)
RecordSet ContainingMatch(Number id, String pattern)
RecordSet ContainingMatch(String name, String pattern)
Returns records containing a match to given pattern in given column. Column can be supplied directly or by its ID or name.

RecordSet ContainingToken(Column column, String token)
RecordSet ContainingToken(Number id, String token)
RecordSet ContainingToken(String name, String token)
Returns records containing given token in given column. Column can be supplied directly or by its ID or name.

Number Correlation(Column column, Column columnSecond)
Number Correlation(Number id, Number idSecond)
Number Correlation(String name, String stringSecond)
Returns correlation of given pair of columns. Columns can be supplied directly or by their IDs or names.

Number Covariation(Column column, Column columnSecond)
Number Covariation(Number id, Number idSecond)
Number Covariation(String name, String stringSecond)
Returns covariation of given pair of columns. Columns can be supplied directly or by their IDs or names.

RecordSet Duplicates(Column column)
RecordSet Duplicates(Number id)
RecordSet Duplicates(String name)
Returns records containing duplicate values of given column. Column can be supplied directly or by its ID or name.

RecordSet DuplicatesExceptFirst(Column column)
RecordSet DuplicatesExceptFirst(Number id)
RecordSet DuplicatesExceptFirst(String name)
Returns record containing duplicate values (except first such value) of given column. Column can be supplied directly or by its ID or name.

RecordSet EndingWith(Column column, String text)
RecordSet EndingWith(Number id, String text)
RecordSet EndingWith(String name, String text)
Returns records whose values in given column end with specified text. Column can be supplied directly or by its ID or name.

RecordSet EndingWithMatch(Column column, String pattern)
RecordSet EndingWithMatch(Number id, String pattern)
RecordSet EndingWithMatch(String name, String pattern)
Returns records whose values in given column end with match to given pattern. Column can be supplied directly or by its ID or name.
RecordSet EndingWithToken(Column column, String token)
RecordSet EndingWithToken(Number id, String token)
RecordSet EndingWithToken(String name, String token)
Returns records whose values in given column end with given token. Column can be supplied directly or by its ID or name.

RecordSet EqualTo(Column column, Variant value)
RecordSet EqualTo(Number id, Variant value)
RecordSet EqualTo(String name, Variant value)
Returns records with values of given column equal to given value. Column can be supplied directly or by its ID or name.

Number Excess(Column column)
Number Excess(Number id)
Number Excess(String name)
Returns excess of given column. Column can be supplied directly or by its ID or name.

RecordSet GreaterOrEqualTo(Column column, Variant value)
RecordSet GreaterOrEqualTo(Number id, Variant value)
RecordSet GreaterOrEqualTo(String name, Variant value)
Returns records with values of given column greater than or equal to given value. Column can be supplied directly or by its ID or name.

RecordSet GreaterThan(Column column, Variant value)
RecordSet GreaterThan(Number id, Variant value)
RecordSet GreaterThan(String name, Variant value)
Returns records with values of given column greater than given value. Column can be supplied directly or by its ID or name.

Number ItemByID(Number id)
Returns index of record with given ID or -1.

Number ItemByValue(Column column, Variant value)
Number ItemByValue(Number id, Variant value)
Number ItemByValue(String name, Variant value)
Returns index of first record with given value of given column or -1. Column can be supplied directly or by its ID or name.

RecordSet LessOrEqualTo(Column column, Variant value)
RecordSet LessOrEqualTo(Number id, Variant value)
RecordSet LessOrEqualTo(String name, Variant value)
Returns records with values of given column less than or equal to given value. Column can be supplied directly or by its ID or name.

RecordSet LessThan(Column column, Variant value)
RecordSet LessThan(Number id, Variant value)
RecordSet LessThan(String name, Variant value)
Returns records with values of given column less than given value. Column can be supplied directly or by its ID or name.

RecordSet Matching(Column column, String pattern)
RecordSet Matching(Number id, String pattern)
RecordSet Matching(String name, String pattern)
Returns records with values of given column matching given pattern. Column can be supplied directly or by its ID or name.

RecordSet Maximum(Column column)
RecordSet Maximum(Number id)
RecordSet Maximum(String name)

Returns RecordSet object with maximum values of given column. Column can be supplied directly or by its ID or name.

Record Median(Column column)
Record Median(Number id)
Record Median(String name)

Returns Record object with median value of given column. Column can be supplied directly or by its ID or name.

RecordSet Minimum(Column column)
RecordSet Minimum(Number id)
RecordSet Minimum(String name)

Returns RecordSet object minimum value of given column. Column can be supplied directly or by its ID or name.

Number Moment(Column column, Number order)
Number Moment(Number id, Number order)
Number Moment(String name, Number order)

Returns momentum of given column. Column can be supplied directly or by its ID or name.

RecordSet NotEqualTo(Column column, Variant value)
RecordSet NotEqualTo(Number id, Variant value)
RecordSet NotEqualTo(String name, Variant value)

Returns records with values of given column not equal to given value. Column can be supplied directly or by its ID or name.

Number Range(Column column)
Number Range(Number id)
Number Range(String name)

Returns range of given column. Column can be supplied directly or by its ID or name.

Remove(Number index)
Remove(Record record)

Removes record from the record set. Removes a record from the table if the record set represents all table records. Unselects the record or removes the record from the saved selection if the record set represents the selection or a saved selection.

RemoveAll()

Removes all records from the record set. Removes all records from the table if the record set represents all table records. Unselects all records or clears the saved selection if the record set represents the selection or a saved selection.

Number Skew(Column column)
Number Skew(Number id)
Number Skew(String name)

Returns skew of given column. Column can be supplied directly or by its ID or name.

RecordSet Sort(Column column)
RecordSet Sort(Number id)
RecordSet Sort(String name)

Returns the same set of records sorted by given column. Column can be supplied directly or by its ID or name.
RecordSet SoundingLike(Column column, String text)
RecordSet SoundingLike(Number id, String text)
RecordSet SoundingLike(String name, String text)
Returns records with values of given column sounding like given text. Column can be supplied directly or by its ID or name.

Number StandardDeviation(Column column)
Number StandardDeviation(Number id)
Number StandardDeviation(String name)
Returns standard deviation of given column. Column can be supplied directly or by its ID or name.

RecordSet StartingWith(Column column, String text)
RecordSet StartingWith(Number id, String text)
RecordSet StartingWith(String name, String text)
Returns records with values of given column starting with given text. Column can be supplied directly or by its ID or name.

RecordSet StartingWithMatch(Column column, String pattern)
RecordSet StartingWithMatch(Number id, String pattern)
RecordSet StartingWithMatch(String name, String pattern)
Returns records with values of given column starting with match to given pattern. Column can be supplied directly or by its ID or name.

RecordSet StartingWithToken(Column column, String token)
RecordSet StartingWithToken(Number id, String token)
RecordSet StartingWithToken(String name, String token)
Returns records with values of given column starting with given token. Column can be supplied directly or by its ID or name.

Number Sum(Column column)
Number Sum(Number id)
Number Sum(String name)
Records sum of given column. Column can be supplied directly or by its ID or name.

RecordSet Top(Column column, Number count)
RecordSet Top(Number id, Number count)
RecordSet Top(String name, Number count)
Records given number of top records for given column. Column can be supplied directly or by its ID or name.

RecordSet Typical(Column column, Number count)
RecordSet Typical(Number id, Number count)
RecordSet Typical(String name, Number count)
Returns given number of typical records for given column. Column can be supplied directly or by its ID or name.

RecordSet Uniques(Column column)
RecordSet Uniques(Number id)
RecordSet Uniques(String name)
Returns records with unique values of given column. Column can be supplied directly or by its ID or name.

Number Variance(Column column)
Number Variance(Number id)
Number Variance(String name)
Returns variance of given column. Column can be supplied directly or by its ID or name.
Notes

Note that RecordSet.Maximum returns another RecordSet object, so to obtain the value of the maximum we would use something along the lines of:

```vbnet
Set recsetMax = recset.Maximum(field)
Set recMax = recsetMax(0)
MsgBox recMax.Data(field)
```

In earlier Manifold releases removing records in the RecordSet object representing all records in a table bound to a drawing would fail with an error message. As of 6.50 SP1, instead of failing this will remove the relevant drawing objects.

Obtained From

- RecordSet (Bottom, Containing, ContainingMatch, ContainingToken, Duplicates, DuplicatesExceptFirst, EndingWith, EndingWithMatch, EndingWithToken, EqualTo, GreaterOrEqualTo, GreaterThan, LessOrEqualTo, LessThan, Matching, Maximum, Median, Minimum, NotEqualTo, Sort, SoundingLike, StartingWith, StartingWithMatch, StartingWithToken, Top, Typical, Uniques)
- Table (RecordSet, Selection)
- TableSelectionSet (Item)

See Also

Scripting Reference
**Rect Object**

Rectangle.

**Properties**

- **Application**
  Returns application. Read only.

- **Number Area**
  Returns rectangle area. Read only.

- **Point Center**
  Sets or returns center of rectangle. Read/write.

- **Number Height**
  Returns rectangle height. Read only.

- **Any Parent**
  Returns parent object. Read only.

- **Number Width**
  Returns rectangle width. Read only.

- **Number XMax**
  Sets or returns maximum X coordinate. Read/write.

- **Number XMin**
  Sets or returns minimum X coordinate. Read/write.

- **Number YMax**
  Sets or returns maximum Y coordinate. Read/write.

- **Number YMin**
  Sets or returns minimum Y coordinate. Read/write.

**Methods**

- **Boolean ContainsPoint(Point point)**
  Returns True if rectangle contains given point and False otherwise.

- **Boolean ContainsRect(Rect rect)**
  Returns True if rectangle contains given rectangle and False otherwise.

- **Copy(Rect rect)**
  Copies given rectangle.

- **Point FitPoint(Point point)**
  Returns a copy of given point constrained by subject rectangle.

- **Rect FitRect(Rect rect)**
  Returns a copy of given rectangle constrained by subject rectangle.
Boolean Intersects(Rect rect)
Returns True if rectangle intersects given rectangle and False otherwise.

Normalize()
Normalizes coordinate pairs so that each minimum coordinate is less than or equal to corresponding maximum coordinate.

UnionPoint(Point point)
Extends rectangle to include given point.

UnionRect(Rect rect)
Extends rectangle to include given rectangle.

**Obtained From**
Application (NewRect)
Branch (Box)
BranchSet (Box)
Geom (Box)
GeomSet (Box)
Label (Box)
LabelSet (Box)
Layout (ScopeArea)
MapServer (DataBox)
ObjectSet (Box)
PixelSet (Box)
PointSet (Box)
Rect (FitRect)

**See Also**

Scripting Reference
Relation Object
Table relation.

Properties

Application Application
Returns application. Read only.

Column Column
Returns source column. Read only.

Any Parent
Returns parent object. Read only.

Column TargetColumn
Returns target column. Read only.

Table Table
Returns source table. Read only.

Table TargetTable
Returns target table. Read only.

Obtained From

RelationSet (Item, LastAdded)

See Also

Scripting Reference
**RelationSet Object**
Set of table relations.

**Properties**

Application Application
Returns application. Read only.

Number Count
Returns number of relations in set. Read only.

Relation Item(Number index)
Returns relation with given index. Read only.

Relation LastAdded
Returns last added relation if any. Read only.

Any Parent
Returns parent object. Read only.

**Methods**

Add(Column column, Column columnTarget)
Appends relation with given source and target columns.

Number ItemByTableID(Number tableID)
Returns index of relation with given table ID or -1.

Remove(Number index)
Removes relation with given index.

RemoveAll()
Removes all relations.

**Obtained From**

Table (RelationSet)

**See Also**

Scripting Reference
RichControl Object
Rich control object.

Properties

Number Height
Sets or returns control height in pixels. Read/write.

Number Left
Sets or returns left coordinate of control in pixels. Read/write.

String Name
Sets or returns control name. Read/write.

String Tag
Sets or returns tag string associated with control. Read/write.

Number Top
Sets or returns top coordinate of control in pixels. Read/write.

Boolean Visible
Sets or returns control visibility state. Read/write.

Number Width
Sets or returns control width in pixels. Read/write.

See Also

Scripting Reference
RichForm Object
Rich form object.

Properties

Color BackColor
Sets or returns background form color. Read/write.

String Caption
Sets or returns form caption. Read/write.

Boolean Centered
Sets or returns form centering option. Read/write.

Boolean ControlBox
Sets or returns form control box option. Read/write.

Boolean Enabled
Sets or returns form enable option. Read/write.

Font Font
Sets or returns form font. Read/write.

Color ForeColor
Sets or returns foreground form color. Read/write.

Number Height
Sets or returns form height in pixels. Read/write.

Number Left
Sets or returns left coordinate of form in pixels. Read/write.

Boolean MaxButton
Sets or returns form maximize button option. Read/write.

Boolean MinButton
Sets or returns form minimize button option. Read/write.

Any Result
Sets or returns form exit code. Read/write.

String Tag
Sets or returns tag string associated with form. Read/write.

Number Top
Sets or returns top coordinate of form in pixels. Read/write.

Boolean Visible
Sets or returns form visibility state. Read/write.

Number Width
Sets or returns form width in pixels. Read/write.
Events

OnActivate()
Occurs when a form becomes the active window.

OnDblClick()
Occurs when the user presses and releases a mouse button and then presses and releases it again over a form.

OnDeactivate()
Occurs when a form is no longer the active window.

OnKeyDown(Number keyCode, Number shift)
Occurs when the user presses a key while a form has the focus.

OnKeyPress(Number keyAscii)
Occurs when the user presses and releases an ANSI key.

OnKeyUp(Number keyCode, Number shift)
Occurs when the user releases a key while a form has the focus.

OnLoad()
Occurs when a form is loaded.

OnMouseDown(Number button, Number shift, Number x, Number y)
Occurs when the user presses the mouse button while a form has the focus.

OnMouseMove(Number button, Number shift, Number x, Number y)
Occurs when the user moves the mouse.

OnMouseUp(Number button, Number shift, Number x, Number y)
Occurs when the user releases the mouse button while a form has the focus.

OnResize()
Occurs when a form is first displayed or the size of an object changes.

OnUnload()
Occurs when a form is about to be removed from the screen.

Comments

Due to potential deadlock issues, it is recommended that one clears the Result property after invoking the form if it is known to contain a reference to an object, like a Component or Application object.

See Also

Scripting Reference
Route
Route object.

Properties

Application Application
Returns application. Read only.

Geom Geom
Returns route metric. Read only.

Any Parent
Returns parent object. Read only.

Methods

AddText(String text)
Appends text to the end of the component.

String Report(Column names, TurnStyle turns)
Reports a route into string. Names column can be supplied directly or by its ID or name. If turn style parameter is omitted, the system uses TurnStyleBoth.

Obtained From

Analyzer (OptimalRoute)

See Also

Scripting Reference
**Script Object**
Script component.

**Properties**

*Application Application*
Returns application. Read only.

*Boolean Cached*
Sets or returns caching option for shared component in Enterprise Edition. Read/write.

*String Description*
Sets or returns component description. Read/write.

*Document Document*
Returns owner document. Read only.

*Folder Folder*
Sets or returns containing folder. Read/write.

*Boolean ForceSeparateThread*
Returns True if script is run in a separate thread and False otherwise, or sets that option. Read / write.

*Number ID*
Returns unique component ID. Read only.

*Boolean IsCheckedOut*
Checks if component is shared and checked out in Enterprise Edition. Read only.

*Boolean IsCheckedOutRemotely*
Checks if component is shared and checked out by another user in Enterprise Edition. Read only.

*Boolean IsModified*
Checks if component is modified since last save in Enterprise Edition. Read only.

*Boolean IsOutdated*
Checks if component is shared and there is a later version of the component on the Enterprise server in Enterprise Edition. Read only.

*Boolean IsReadOnly*
Checks if component is readonly (eg, shared and not checked out) in Enterprise Edition. Read only.

*Boolean IsShared*
Checks if component is shared on an Enterprise server in Enterprise Edition. Read only.

*Boolean IsWritable*
Checks if component is writable in Enterprise Edition. Read only.

*String Language*
Sets or returns name of script engine used to run the script (eg. "VBScript"). Read only.

*String LanguageProgID*
Sets or returns ProgID of script engine used to run the script. Read only.
Number Length
Returns number of characters in script text. Read only.

String Name
Sets or returns component name. Read/write.

String Note
Sets or returns notes associated with the component. Read/write.

ComponentSet OwnedComponentSet
Returns set of components bound to this component. Read only.

ComponentSet OwnedLayoutSet
Returns set of layouts bound to this component. Read only.

Component Owner
Returns owner component if any. Read only.

Any Parent
Returns parent object. Read only.

Number RemoteVersion
Returns version of shared component available on the Enterprise server in Enterprise Edition. Read only.

String Server
Returns server identification string for shared component in Enterprise Edition. Read only.

String ShareStatus
Returns textual report on component sharing status in Enterprise Edition. Read only.

String Text
Sets or returns script text. Read/write.

ComponentType Type
Returns component type. Read only.

String TypeName
Returns textual representation of the component type. Read only.

Number Version
Returns version of shared component within the local project in Enterprise Edition. Read only.

Methods

AddText(String text)
Appends text to the end of the component.

CheckIn()
Checks in changes made to a shared component to the Enterprise server in Enterprise Edition.

CheckOut()
Checks out a shared component for editing in Enterprise Edition.
Clear()
Clears component.

Copy()
Copies component into the Clipboard.

Cut()
Copies component into the Clipboard and then clears it.

GetLatestVersion()
Gets the latest version of a shared component from the Enterprise server in Enterprise Edition.

Boolean IsEmpty(Boolean ignoreWhiteSpaces)
Returns True if component text is empty and False otherwise.

Open()
Opens component.

OpenInNewWindow()
Opens component in new window.

Paste(Boolean replace)
Pastes the contents of the Clipboard into the component replacing previous data or adding to it. Calling the method without parameters replaces data.

Print(Boolean useDialog)
Prints component.

Run()
Runs script.

UndoCheckOut()
Undoes changes made to a shared component and gets the latest version of the component from the Enterprise server in Enterprise Edition.

Obtained From
Document (NewScript)

See Also
Scripting Reference
**Surface Object**
Surface component.

**Properties**

Application Application
Returns application. Read only.

Color BackgroundColor
Sets or returns background color. Read / write.

Rect Box
Returns bounding box of all pixels. Read only.

Boolean Cached
Sets or returns caching option for shared component in Enterprise Edition. Read/write.

ControlPointSet ControlPointSet
Returns set of control points. Read only.

CoordinateSystem CoordinateSystem
Sets or returns coordinate system. Read/write.

Boolean CoordinateSystemVerified
Checks whether or not the component coordinate system has been verified by the user, or alters the verification state. Read / write.

View DefaultView
Returns default view. Read only.

String Description
Sets or returns component description. Read/write.

Document Document
Returns owner document. Read only.

Folder Folder
Sets or returns containing folder. Read/write.

Number Height
Returns surface height in pixels. Read only.

Number ID
Returns unique component ID. Read only.

Boolean IsCheckedOut
Checks if component is shared and checked out in Enterprise Edition. Read only.

Boolean IsCheckedOutRemotely
Checks if component is shared and checked out by another user in Enterprise Edition. Read only.

Boolean IsModified
Checks if component is modified since last save in Enterprise Edition. Read only.
Boolean IsOutdated
Checks if component is shared and there is a later version of the component on the Enterprise server in Enterprise Edition. Read only.

Boolean IsReadOnly
Checks if component is readonly (eg, shared and not checked out) in Enterprise Edition. Read only.

Boolean IsShared
Checks if component is shared on an Enterprise server in Enterprise Edition. Read only.

Boolean IsWritable
Checks if component is writable in Enterprise Edition. Read only.

String LinkRowset
Returns the source rowset of a linked component. Read only.

String LinkSource
Returns the data source of a linked component. Read only.

String LinkTechnology
Returns the data access technology used by a linked component. Read only.

String Name
Sets or returns component name. Read/write.

String Note
Sets or returns notes associated with the component. Read/write.

ComponentSet OwnedComponentSet
Returns set of components bound to this component. Read only.

ComponentSet OwnedLayoutSet
Returns set of layouts bound to this component. Read only.

ComponentSet OwnedProfileSet
Returns set of profiles bound to this component. Read only.

ComponentSet OwnedTerrainSet
Returns set of terrains bound to this component. Read only.

Component Owner
Returns owner component if any. Read only.

Any Parent
Returns parent object. Read only.

PixelSet PixelSet
Returns set of surface pixels. Read only.

Number RemoteVersion
Returns version of shared component available on the Enterprise server in Enterprise Edition. Read only.
SurfaceSelectionSet SavedSelectionSet
Returns set of saved selections. Read only.

PixelSet Selection
Returns set of selected pixels. Read only.

String Server
Returns server identification string for shared component in Enterprise Edition. Read only.

String ShareStatus
Returns textual report on component sharing status in Enterprise Edition. Read only.

ComponentType Type
Returns component type. Read only.

String TypeName
Returns textual representation of the component type. Read only.

ValueType ValueType
Returns type of surface pixel values. Read only.

Number Version
Returns version of shared component within the local project in Enterprise Edition. Read only.

ViewSet ViewSet
Returns set of views associated with this component. Read only.

Number Width
Returns surface width in pixels. Read only.

Number ZoomMax
Sets or returns maximum zoom level. Read/write.

Number ZoomMin
Sets or returns minimum zoom level. Read/write.

Methods

CheckIn()
Checks in changes made to a shared component to the Enterprise server in Enterprise Edition.

CheckOut()
Checks out a shared component for editing in Enterprise Edition.

Clear(Boolean selection)
Clears entire component or selection within the component. Calling the method without parameters clears entire component.

Copy(Boolean selection)
Copies entire component or selection within the component into the Clipboard. Calling the method without parameters copies entire component.

CopyBitmap(Boolean selection)
Copies entire component or selection within the component into the Clipboard as a bitmap. Calling the method without parameters copies entire component.

Cut(Boolean selection)
Copies entire component or selection within the component into the Clipboard and then clears it. Calling the method without parameters cuts entire component.

CutBitmap(Boolean selection)
Copies entire component or selection within the component into the Clipboard as a bitmap and then clears it. Calling the method without parameters cuts entire component.

GetLatestVersion()
Gets the latest version of a shared component from the Enterprise server in Enterprise Edition.

Boolean IsLinked()
Returns True if surface is linked and False otherwise.

Open()
Opens component.

OpenInNewWindow()
Opens component in new window.

Paste(Boolean replace)
Pastes the contents of the Clipboard into the component replacing current selection or leaving it intact. Calling the method without parameters replaces selection.

ProjectTo(CoordinateSystem system, Boolean adjustResolution)
Projects component to another coordinate system. If the second parameter is True or omitted, the system automatically adjusts the local scale parameters of the coordinate system to match the current resolution of the image.

Print(Boolean useDialog)
Prints component.

RenderAreaTo(String name, Number width, Number height, Rect area, Boolean clean)
Renders component area into an image with given name, width and height. If last parameter is True, legend, north arrow and scale bar are not rendered even if they are on.

RenderTo(String name, Number width, Number height, Boolean clean)
Render component into an image with given name, width and height. If last parameter is True, legend, north arrow and scale bar are not rendered even if they are turned on.

SelectAll()
Selects all pixels.

SelectInverse()
Selects pixels that were not selected and vice versa.

SelectNone()
Unselects all pixels.

UndoCheckOut()
Undoes changes made to a shared component and gets the latest version of the component from the Enterprise server in Enterprise Edition.
Unlink()
Unlinks component. If the component is not linked, does nothing.

Obtained From
Document (NewSurface)

See Also
Scripting Reference
**SurfaceSelectionSet Object**
Set of saved selections within surface.

**Properties**

- **Application Application**
  Returns application. Read only.

- **Number Count**
  Returns number of selections in set. Read only.

- **PixelSet Item(Number index)**
  Returns selection with given index. Read only.

- **Any Parent**
  Returns parent object. Read only.

**Methods**

- **Number ItemByMask(Number mask)**
  Returns index of selection with given mask value or -1.

- **Number ItemByName(String name)**
  Returns index of selection with given name or -1.

**Obtained From**

- Surface (SavedSelectionSet)

**See Also**

Scripting Reference
Table Object
Table component.

Properties

Application Application
Returns application. Read only.

Boolean Cached
Sets or returns caching option for shared component in Enterprise Edition. Read/write.

ColumnSet ColumnSet
Returns set of table columns. Read only.

String Description
Sets or returns component description. Read/write.

Document Document
Returns owner document. Read only.

DSSAtomSet DSSAtomSet
Returns set of atomic expressions used in rank columns. Read only.

Folder Folder
Sets or returns containing folder. Read/write.

Number ID
Returns unique component ID. Read only.

Column Identity
Returns column used as identity. Read only.

Boolean IsCheckedOut
Checks if component is shared and checked out in Enterprise Edition. Read only.

Boolean IsCheckedOutRemotely
Checks if component is shared and checked out by another user in Enterprise Edition. Read only.

Boolean IsModified
Checks if component is modified since last save in Enterprise Edition. Read only.

Boolean IsOutdated
Checks if component is shared and there is a later version of the component on the Enterprise server in Enterprise Edition. Read only.

Boolean IsReadOnly
Checks if component is readonly (eg, shared and not checked out) in Enterprise Edition. Read only.

Boolean IsShared
Checks if component is shared on an Enterprise server in Enterprise Edition. Read only.

Boolean IsWritable
Checks if component is writable in Enterprise Edition. Read only.
DateTime LastRefreshed
Returns date and time of the last refresh for a linked component and the current date and time for a non-linked component.

String LinkRowset
Returns the source rowset of a linked component. Read only.

String LinkSource
Returns the data source of a linked component. Read only.

String LinkTechnology
Returns the data access technology used by a linked component. Read only.

String Name
Sets or returns component name. Read/write.

String Note
Sets or returns notes associated with the component. Read/write.

ComponentSet OwnedComponentSet
Returns set of components bound to this component. Read only.

ComponentSet OwnedLayoutSet
Returns set of layouts bound to this component. Read only.

Component Owner
Returns owner component if any. Read only.

Any Parent
Returns parent object. Read only.

RecordSet RecordSet
Returns set of table records. Read only.

RelationSet RelationSet
Returns set of table relations. Read only.

Number RemoteVersion
Returns version of shared component available on the Enterprise server in Enterprise Edition. Read only.

TableSelectionSet SavedSelectionSet
Returns set of saved selections. Read only.

RecordSet Selection
Returns set of selected records. Read only.

String Server
Returns server identification string for shared component in Enterprise Edition. Read only.

String ShareStatus
Returns textual report on component sharing status in Enterprise Edition. Read only.
ComponentType Type
Returns component type. Read only.

String TypeName
Returns textual representation of the component type. Read only.

Unlink()
Unlinks component. If the component is not linked, does nothing.

Number Version
Returns version of shared component within the local project in Enterprise Edition. Read only.

Methods

CheckIn()
Checks in changes made to a shared component to the Enterprise server in Enterprise Edition.

CheckOut()
Checks out a shared component for editing in Enterprise Edition.

Clear(Boolean selection)
Clears entire component or selection within the component. Calling the method without parameters clears entire component.

Copy(Boolean selection)
Copies entire component or selection within the component into the Clipboard. Calling the method without parameters copies entire component.

Cut(Boolean selection)
Copies entire component or selection within the component into the Clipboard and then clears it. Calling the method without parameters cuts entire component.

GetLatestVersion()
Gets the latest version of a shared component from the Enterprise server in Enterprise Edition.

Boolean IsLinked()
Returns True if table is linked and False otherwise.

Boolean IsNative()
Returns True if table is native and False otherwise.

Boolean IsQuery()
Returns True if table is computed by the query and False otherwise.

Open()
Opens component.

OpenInNewWindow()
Opens component in new window.

Paste(Boolean replace)
Pastes the contents of the Clipboard into the component replacing current selection or leaving it intact. Calling the method without parameters replaces selection.
Print(Boolean useDialog)
Prints component.

Boolean Refresh()
Refreshes a linked component and does nothing for a non-linked component. In the case of a linked component returns True if the refresh has succeeded and False otherwise. Refreshing a table bound to a linked drawing refreshes the drawing and vice versa.

Boolean RefreshAfter(Number)
Refreshes a linked component if the component has not been refreshed for more than the given number of seconds.

SelectAll()
Selects all records.

SelectInverse()
Selects records that were not selected and vice versa.

SelectNone()
Unselects all records.

UndoCheckOut()
Undoes changes made to a shared component and gets the latest version of the component from the Enterprise server in Enterprise Edition.

Obtained From
- Column (Table)
- Document (NewTable)
- Drawing (OwnedTable)
- Query (Table)
- Relation (Table, TargetTable)

See Also
Scripting Reference
**TableSelectionSet Object**
Set of saved selections within table.

**Properties**

Application Application
Returns application. Read only.

Number Count
Returns number of selections in set. Read only.

RecordSet Item(Number index)
Returns selection with given index. Read only.

Any Parent
Returns parent object. Read only.

**Methods**

Number ItemByMask(Number mask)
Returns index of selection with given mask value or -1.

Number ItemByName(String name)
Returns index of selection with given name or -1.

**Obtained From**

Table (SavedSelectionSet)

**See Also**

Scripting Reference
**TableWindow Object**

Table window.

**Properties**

- **Column ActiveColumn**
  Returns active column. Read only.

- **Component ActiveComponent**
  Returns table component shown in window. Read only.

- **Record ActiveRecord**
  Records active record. Read only.

- **Application Application**
  Returns application. Read only.

- **Component Component**
  Returns table component shown in window. Read only.

- **Number Height**
  Returns window height in pixels. Read only.

- **Any Parent**
  Returns parent object. Read only.

- **Number Width**
  Returns window width in pixels. Read only.

**Methods**

- **Close()**
  Closes window.

- **Boolean HasActiveRecord()**
  Returns True if window focus is set on some record and False otherwise.

- **Boolean HasScaleInternal()**
  Returns False.

- **Refresh()**
  Updates window contents.

**Obtained From**

- **WindowSet (Item, ActiveWindow)**

**See Also**

- Scripting Reference
**Terrain Object**
Terrain component.

**Properties**

**Application Application**
Returns application. Read only.

**Boolean Areas**
Toggles vector shapes for area objects. Read/write.

**Boolean AreaHeightColumn**
Toggles the use of a height column for area objects. Applies to vector shapes. Setting the property to True retrieves the height of each area object relative to the surface from the column named "Height". Setting the property to False sets the height of each area object relative to the surface to 0. Read/write.

**Number AreaOpacity**
Sets or returns opacity of area objects. Applies to vector shapes. Read/write.

**Boolean AreasTex**
Toggles embedding of area objects into textures. Read/write.

**Boolean AreaWalls**
Toggles vertical walls around area objects. Applies to vector shapes. Read/write.

**Color BackgroundColor**
Sets or returns background color. Read/write.

**CameraType CameraType**
Sets or returns camera type. Read/write.

**Number CloudAlpha (formerly named CloudsAlpha)**
Sets or returns cloud transparency. Read/write.

**Number CloudCount**
Sets or returns the number of clouds. Read/write.

**Number CloudHeight**
Sets or returns cloud height. Read/write.

**Boolean Clouds**
Toggles the rendering of clouds. Read/write.

**Number CloudSize**
Sets or returns cloud size in pixels. Read/write.

**String Description**
Sets or returns component description. Read/write.

**Document Document**
Returns owner document. Read only.

**Boolean Fog**
Toggles fog. Read/write.

Color FogColor
Sets or returns fog color. Read/write.

Number FogFar
Sets or returns the distance to the farthest fog clipping plane. Read/write.

Number FogNear
Sets or returns the distance to the nearest fog clipping plane. Read/write.

Folder Folder
Sets or returns containing folder. Read/write.

Number ID
Returns unique component ID. Read only.

Boolean LabelBase
Toggles base (pier) lines for label objects. Applies to vector shapes. Read/write.

Number LabelHeight
Sets or returns height of label objects. Applies to vector shapes. Read/write.

Boolean Labels
Toggles vector shapes for label objects. Read/write.

Number LabelSize
Sets or returns size of label objects. Applies to vector shapes. Read/write.

Number LabelStep
Sets or returns spacing between curved label objects in pixels. Applies to vector shapes. Read/write.

Boolean LabelsTex
Toggles embedding of label objects into textures. Read/write.

Boolean Legend
Toggles legend. Read/write.

WindowAlign LegendAlign
Sets or returns legend alignment. Read/write.

String LegendCaption
Sets or returns legend caption. Read/write.

LegendStyle LegendStyle (formerly named LegendType)
Sets or returns legend style. Read/write.

Boolean Lighting
Toggles lighting. Read/write.

Number LightingAlpha
Sets or returns lighting transparency. Read/write.
Programming Reference

LightingDirection
Sets or returns lighting direction. Read/write.

Number LightingDirX
Returns X component of lighting direction (-1, 0 or 1). Read only.

Number LightingDirY
Returns Y component of lighting direction (-1, 0 or 1). Read only.

Boolean LineBase
Toggles base (pier) lines for line objects. Applies to vector shapes. Read/write.

Number LineHeight
Sets or returns height of line objects. Applies to vector shapes. Read/write.

Boolean Lines
Toggles vector shapes for line objects. Read/write.

Number LineSize
Sets or returns size of line objects. Applies to vector shapes. Read/write.

Boolean LineSmooth
Toggles antialiasing of line objects. Applies to vector shapes. Read/write.

Boolean LinesTex
Toggles embedding of line objects into textures. Read/write.

String Name
Sets or returns component name. Read/write.

String Note
Sets or returns notes associated with the component. Read/write.

Boolean NorthArrow
Toggles north arrow. Read/write.

WindowAlign NorthArrowAlign
Sets or returns north arrow alignment. Read/write.

Color NorthArrowBackgroundColor (formerly named NorthArrowBackColor)
Sets or returns north arrow background color. Read/write.

Color NorthArrowColor (formerly named NorthArrowForeColor)
Sets or returns north arrow foreground color. Read/write.

Number NorthArrowSize
Sets or returns north arrow size in points. Read/write.

NorthArrowStyle NorthArrowStyle (formerly named NorthArrowType)
Sets or returns north arrow style. Read/write.

ComponentSet Overlay
Returns set of overlaid components. Read only.
ComponentSet OwnedComponentSet
Returns set of components bound to this component. Read only.

Component Owner
Returns owner component if any. Read only.

Any Parent
Returns parent object. Read only.

Boolean PointFrame
Toggles rectangular frames around point objects. Applies to vector shapes. Read/write.

Number PointHeight
Sets or returns height of point objects. Applies to vector shapes. Read/write.

Boolean Points
Toggles vector shapes for point objects. Read/write.

Number PointSize
Sets or returns size of point objects. Read/write.

Number PointStyle
Sets or returns style of point objects. Applies to vector shapes. Read/write.

Boolean PointsTex
Toggles embedding of point objects into textures. Read/write.

Number Quality
Sets or returns surface rendering quality. The less the number, the greater the quality with the value of 0 meaning the best quality possible. Read/write.

Number TextureOpacity
Sets or returns texture opacity. Read/write.

ComponentType Type
Returns component type. Read only.

String TypeName
Returns textual representation of the component type. Read only.

Number VisibleTiles
Sets or returns the number of visible tiles from the camera. Read/write.

Boolean Walls
Toggles vertical walls around patches of invisible pixels within the surface and at the edges of the surface. Read/write.

Boolean Water
Toggles water. Read/write.

Number WaterAlpha
Sets or returns water transparency. Read/write.
Color WaterColor
Sets or returns water color. Read/write.

Number WaterFog
Sets or returns water fog value. Read/write.

Number WaterLevel
Sets or returns water level. Read/write.

Number WaterQuality
Sets or returns water rendering quality. Read/write.

Boolean Wireframe
Toggles wireframe rendering mode. Read/write.

Number ZScale
Sets or returns Z scale factor. Read/write.

Methods

Copy()
Copies component into the Clipboard.

Open()
Opens component.

OpenInNewWindow()
Opens component in new window.

Paste()
Pastes the contents of the Clipboard into the component replacing previous data.

Print(Boolean useDialog)
Prints component.

RenderTo(String name, Number x, Number y, Number cameraX, Number cameraY, Number cameraZ, Number cameraDX, Number cameraDY, Number cameraDZ)
Render component into an image with given name, width and height, using given location and direction of the camera. The last three parameters which give the direction of the camera can be omitted.

Obtained From

Document (NewTerrain)

See Also

Scripting Reference
TerrainWindow Object
Terrain window.

Properties

Component ActiveComponent
Returns terrain component shown in window. Read only.

Application Application
Returns application. Read only.

Number CameraDX
Sets or returns X direction of a camera. Read/write.

Number CameraDY
Sets or returns Y direction of a camera. Read/write.

Number CameraDZ
Sets or returns Z direction of a camera. Read/write.

Number CameraHeight
Sets or returns camera height. Read/write.

Component Component
Returns terrain component shown in window. Read only.

Number Height
Returns window height in pixels. Read only.

Point Location
Sets or returns viewport center. Read/write.

Point LocationLatLon
Sets or returns viewport center in lat/lon coordinates. Read/write.

Any Parent
Returns parent object. Read only.

Number SurfaceHeight
Returns height of surface at current camera location. Read only.

Number Width
Returns window width in pixels. Read only.

Methods

Close()
Closes window.

Boolean HasLocation()
Returns True.
Boolean HasScale()
  Returns False.

Boolean HasScaleInternal()
  Returns False.

MoveTo(Branch branch)
MoveTo(BranchSet branchSet)
MoveTo(Component component)
MoveTo(Geom geom)
MoveTo(Label label)
MoveTo(LabelSet labelSet)
MoveTo(Object object)
MoveTo(ObjectSet objectSet)
MoveTo(Pixel pixel)
MoveTo(PixelSet pixelSet)
MoveTo(Point point)
MoveTo(PointSet pointSet)
MoveTo(Rect rect)
  Centers viewport on given object or set of objects. Branches, geoms, points, rects and their sets are assumed to be in the coordinate system of the displayed component.

MoveToLocation(Point point, Boolean latLon)
  Centers viewport on given location. The last parameter is optional.

Refresh()
  Updates window contents.

RenderTo(String name)
  Renders window into an image with given name.

Obtained From

WindowSet (Item, ActiveWindow)

See Also

Scripting Reference
Theme Object
Theme component.

Properties

Application Application
Returns application. Read only.

Format AreaBackground
Returns background color used for areas. Read only.

Format AreaBorderBackground
Returns border background color used for areas. Read only.

Format AreaBorderForeground
Returns border foreground color used for areas. Read only.

Format AreaBorderSize
Returns border size used for areas. Read only.

Format AreaBorderStyle
Returns border style used for areas. Read only.

Format AreaForeground
Returns foreground color used for areas. Read only.

Format AreaSize
Returns size used for areas. Read only.

Format AreaStyle
Returns style used for areas. Read only.

Color BackgroundColor
Sets or returns background color. Read / write.

Boolean Cached
Sets or returns caching option for shared component in Enterprise Edition. Read/write.

ControlPointSet ControlPointSet
Returns set of control points of parent drawing. Read only.

CoordinateSystem CoordinateSystem
Sets or returns coordinate system of parent drawing. Read/write.

Boolean CoordinateSystemVerified
Checks whether or not the component coordinate system has been verified by the user, or alters the verification state. Read / write.

View DefaultView
Returns default view of parent drawing. Read only.

String Description
Sets or returns component description. Read/write.
Document
  Returns owner document. Read only.

Number Epsilon
  Sets or returns location precision of parent drawing. Read/write.

Folder Folder
  Sets or returns containing folder. Read/write.

Number ID
  Returns unique component ID. Read only.

Boolean IsCheckedOut
  Checks if component is shared and checked out in Enterprise Edition. Read only.

Boolean IsCheckedOutRemotely
  Checks if component is shared and checked out by another user in Enterprise Edition. Read only.

Boolean IsModified
  Checks if component is modified since last save in Enterprise Edition. Read only.

Boolean IsOutdated
  Checks if component is shared and there is a later version of the component on the Enterprise server in Enterprise Edition. Read only.

Boolean IsReadOnly
  Checks if component is readonly (eg, shared and not checked out) in Enterprise Edition. Read only.

Boolean IsShared
  Checks if component is shared on an Enterprise server in Enterprise Edition. Read only.

Boolean IsWritable
  Checks if component is writable in Enterprise Edition. Read only.

DateTime LastRefreshed
  Returns date and time of the last refresh for a linked parent drawing and the current date and time for a non-linked parent drawing.

Format LineBackground
  Returns background color used for lines. Read only.

Format LineForeground
  Returns foreground color used for lines. Read only.

Format LineSize
  Returns size used for lines. Read only.

Format LineStyle
  Returns style used for lines. Read only.

String Name
  Sets or returns component name. Read/write.
String Note
Sets or returns notes associated with the component. Read/write.

ObjectSet ObjectSet
Returns set of objects in parent drawing. Read only.

ComponentSet OwnedComponentSet
Returns set of components bound to this component. Read only.

ComponentSet OwnedLayoutSet
Returns set of layouts bound to this component. Read only.

Drawing OwnerDrawing
Returns parent drawing. Read only.

Component Owner
Returns parent drawing. Read only.

Any Parent
Returns parent object. Read only.

Format PointBackground
Returns background color used for points. Read only.

Format PointForeground
Returns foreground color used for points. Read only.

Format PointRotation
Returns rotation used for points. Read only.

Format PointSize
Returns size used for points. Read only.

Format PointStyle
Returns style used for points. Read only.

Number RemoteVersion
Returns version of shared component available on the Enterprise server in Enterprise Edition. Read only.

DrawingSelectionSet SavedSelectionSet
Returns set of saved selections in parent drawing. Read only.

ObjectSet Selection
Returns set of selected objects in parent drawing. Read only.

String Server
Returns server identification string for shared component in Enterprise Edition. Read only.

String ShareStatus
Returns textual report on component sharing status in Enterprise Edition. Read only.

ComponentType Type
Returns component type. Read only.

String TypeName
Returns textual representation of the component type. Read only.

Number Version
Returns version of shared component within the local project in Enterprise Edition. Read only.

ViewSet ViewSet
Returns set of views associated with parent drawing. Read only.

Number ZoomMax
Sets or returns maximum zoom level. Read/write.

Number ZoomMin
Sets or returns minimum zoom level. Read/write.

Number ZoomRender
Sets or returns render zoom level. Read/write.

Methods

CheckIn()
Checks in changes made to a shared component to the Enterprise server in Enterprise Edition.

CheckOut()
Checks out a shared component for editing in Enterprise Edition.

Clear(Boolean selection)
Clears entire component or selection within the component. Calling the method without parameters clears entire component.

Copy(Boolean selection)
Copies entire component or selection within the component into the Clipboard. Calling the method without parameters copies entire component.

Cut(Boolean selection)
Copies entire component or selection within the component into the Clipboard and then clears it. Calling the method without parameters cuts entire component.

GetLatestVersion()
Gets the latest version of a shared component from the Enterprise server in Enterprise Edition.

Boolean IsEmpty()
Returns True if parent drawing contains no objects and False otherwise.

Boolean IsLinked()
Returns True if parent drawing is linked and False otherwise.

Open()
Opens component.

OpenInNewWindow()
Opens component in new window.
Paste(Boolean replace)
  Pastes the contents of the Clipboard into the component replacing current selection or leaving it intact. Calling the method without parameters replaces selection.

ProjectTo(CoordinateSystem system)
  Projects parent drawing to another coordinate system.

Print(Boolean useDialog)
  Prints component.

RenderAreaTo(String name, Number width, Number height, Rect area, Boolean clean)
  Renders component area into an image with given name, width and height. If last parameter is True, legend, north arrow and scale bar are not rendered even if they are on.

RenderTo(String name, Number width, Number height, Boolean clean)
  Render component into an image with given name, width and height. If last parameter is True, legend, north arrow and scale bar are not rendered even if they are turned on.

Boolean Refresh()
  Refreshes a linked parent drawing and does nothing for a non-linked parent drawing. In the case of a linked parent drawing returns True if the refresh has succeeded and False otherwise. Refreshing a table bound to a linked drawing refreshes the drawing and vice versa.

Boolean RefreshAfter(Number)
  Refreshes a linked parent drawing if it has not been refreshed for more than the given number of seconds.

UndoCheckOut()
  Undoes changes made to a shared component and gets the latest version of the component from the Enterprise server in Enterprise Edition.

Obtained From

Document (NewTheme)

See Also

Scripting Reference
**Topology Object**
Topology data. To use the object, first bind it to a particular drawing using the Bind method.

**Properties**

Application Application
Returns application. Read only.

Drawing Drawing
Returns bound drawing. Read only.

Number Epsilon
Returns location precision of the bound drawing at the time topology data was created. Read only.

Any Parent
Returns parent object. Read only.

Number Version
Returns a numeric value representing the version of the geometry in the bound drawing at the time topology data was created. Read only.

**Methods**

Bind(Drawing drawing)
Binds topology object to drawing.

Build()
Builds topology for the bound drawing. If the drawing contains areas, builds topology for areas and ignores lines and points. If the drawing contains no areas but contains lines, builds topology for lines and ignores points.

DoIdentity(Topology overlay, String name)
Performs an "identity" topology overlay and saves resulting data to a new drawing. Overlaid topology should contain data for areas.

DoIntersect(Topology overlay, String name)
Performs an "intersect" topology overlay and saves resulting data to a new drawing. Both topology objects should contain data for areas.

DoUnion(Topology overlay, String name)
Performs a "union" topology overlay and saves resulting data to a new drawing. Both topology objects should contain data for areas.

DoUpdate(Topology overlay, String name)
Performs an "update" topology overlay and saves resulting data to a new drawing. Both topology objects should contain data for areas.

Boolean IsCompatibleWith(Topology overlay)
Returns True if performing a topology overlay with given topology object will use topology data cached in that object, and False if performing a topology overlay will rebuild topology from scratch. Two topology objects are only compatible with each other if their drawings are in the same coordinate system and use the same location precision.

Boolean IsUpToDate(String filename)
Returns True if topology stored in given file is valid and up to date with the current version of the bound drawing and false otherwise.
Load(String filename)
   Loads topology data from given file.

Output(String name)
   Rebuilds objects in the bound drawing using created topology data and saves resulting objects to a new
drawing.

Save(String filename)
   Saves topology data to given file. If the file already exists, it will be overwritten.

Obtained From

Application (NewTopology)

See Also

Scripting Reference
**TriangleSet Object**
Set of triangles. Each triangle is identified by three vertex indices.

**Properties**

**Application Application**
Returns application. Read only.

**Array Indices**
Returns an array of vertex indices. Elements 0, 1 and 2 identify vertices for the first triangle, elements 3, 4 and 5 identify vertices for the second triangle, and so on. Read only.

**Any Parent**
Returns parent object. Read only.

**Array Xs**
Returns an array of X coordinates for vertices. Read only.

**Array Ys**
Returns an array of Y coordinates for vertices. Read only.

**Obtained From**

**Geom** (DecomposeToTrianglesAdv)

**See Also**

Scripting Reference
Unit Object
Coordinate system unit.

Properties

Application
Application
Returns application. Read only.

String Name
Returns unit name. Read only.

String NickName
Returns unit nick name. Read only.

String NickNameArea
Returns unit nick name for area measurements. Read only.

Any Parent
Returns parent object. Read only.

Number Scale
Returns unit scale with respect to degree or meter. Read only.

Methods

Boolean IsLatLon()
Returns True for angular units and False for metric units.

Load(String name)
Loads unit using given name.

Obtained From

Application (DefaultUnit, DefaultUnitLatLon, NewUnit)
LayoutEntry (BorderUnit)
UnitSet (Item)

See Also

Scripting Reference
UnitSet Object
Set of units.

Properties

Application Application
Returns application. Read only.

Number Count
Returns number of units in set. Read only.

Unit Item(Number index)
Unit Item(String name)
Returns unit with given index or name. Read only.

Any Parent
Returns parent object. Read only.

Methods

Number ItemByAnyName(String name)
Returns index of unit with given name, nick name or area nick name, or -1.

Number ItemByName(String name)
Returns index of unit with given name or -1.

Number ItemByNickName(String name)
Returns index of unit with given nick name or -1.

Number ItemByNickNameArea(String name)
Returns index of unit with given area nick name or -1.

Obtained From

Application (UnitSet)

See Also

Scripting Reference
**UserInterface Object**

User interface manager. Allows invoking commands and accessing user interface elements such as dialogs and controls. Scripting the user interface requires that the script is run in a separate thread.

**Properties**

- **Application**
  - Returns application. Read only.

- **Boolean DisplaysModalDialog**
  - Returns True if user interface displays a modal dialog. Read only.

- **UserInterfaceDialog ModalDialog**
  - Returns topmost modal dialog. Read only.

- **Any Parent**
  - Returns parent object. Read only.

- **UserInterfaceDialogSet Panes**
  - Returns panes. Panes can be accessed by names. Read only.

- **UserInterfaceDialogSet Toolbars**
  - Returns toolbars owned by the main window. Individual toolbars can be accessed by names. Read only.

**Methods**

- **InvokeCommand(String name, String context)**
  - Invokes command with given name. To see the names of available commands, launch Manifold from the command line specifying the "clist" command line option. The context parameter is optional. If supplied, it uses one of the following patterns:

  - "cell:<rowNumber>:<columnName>" - executes command in the context of the table cell in the specified column and row in the active table window,

  - "column:<columnName>" - executes command in the context of the specified column in the active table window,

  - "element:<elementID>" - executes command in the context of the specified layout element in the active layout window,

  - "label:<labelID>" - executes command in the context of the specified label in the active labels window or map window,

  - "layer:<layerName>" - executes command in the context of the specified layer in the active map window,

  - "object:<objectId>" - executes command in the context of the specified drawing object in the active drawing window or map window,

  - "pane:<paneName>" - executes command in the context of the specified pane,

  - "row:<rowNumber>" - executes command in the context of the specified row in the active table window.

**Obtained From**

- Application (UserInterface)
See Also

Scripting Reference
UserInterfaceControl Object
A user interface control, for example, a button or a combo box. A control can contain items.

Properties

Application Application
- Returns application. Read only.

Boolean Checked
- Sets or returns the check state. Read / write.

Boolean Enabled
- Returns True if control is enabled and False otherwise. Read only.

UserInterfaceControlItemSet ItemSet
- Returns control items. Read only.

String Name
- Returns control name. Read only.

Any Parent
- Returns parent object. Read only.

String Text
- Sets or returns control text. Read / write.

Method

Push()
- Simulates clicking on a push button.

PushNamed(String name)
- Push the specified button in the drop-down portion of the control. Used with color, size, and style wells whose drop-down portions contain buttons for thematic formatting ("Theme") and with similar windows.

Obtained From

UserInterfaceControlSet (Item)

See Also

Scripting Reference
UserInterfaceControlItem Object
An item within a user interface control, for example, in a list view or a tree view. An item can contain other items.

Properties

Application Application
Returns application. Read only.

Boolean Checked
Sets or returns the check state. Read / write.

Any Parent
Returns parent object. Read only.

UserInterfaceControlItemSet SubitemSet
Returns subitems. Read only.

String Text
Returns item text. Read only.

String Value
Gets or sets item value. Read / write.

Obtained From

UserInterfaceControlItemSet (Item)

See Also

Scripting Reference
**UserInterfaceControlItemSet Object**
Set of items within a user interface control, for example, a list view or a tree view.

**Properties**

Application Application
Returns application. Read only.

Number Count
Returns number of items in set. Read only.

UserInterfaceControlItem Item(Number index)  UserInterfaceControlItem Item(String text)
Returns item with given index or text. Read only.

Any Parent
Returns parent object. Read only.

**Methods**

Number ItemByName(String text)
Returns index of item with given text or -1.

**Obtained From**

UserInterfaceControl (ItemSet)
UserInterfaceControlItem (SubitemSet)

**See Also**

Scripting Reference
UserInterfaceControlSet Object
Set of user interface controls, for example, in a dialog.

Properties

Application Application
Returns application. Read only.

Number Count
Returns number of controls in set. Read only.

UserInterfaceControl Item(Number index) UserInterfaceControl Item(String name)
Returns control with given index or name. Read only.

Any Parent
Returns parent object. Read only.

Methods

Number ItemBy_Name(String name)
Returns index of control with given name or -1.

Obtained From

UserInterfaceDialog (ControlSet)

See Also

Scripting Reference
UserInterfaceDialog Object
User interface dialog.

Properties

Application Application
Returns application. Read only.

String Caption
Returns dialog caption. Read only.

UserInterfaceControlSet ControlSet
Returns dialog controls. Read only.

Any Parent
Returns parent object. Read only.

Methods

Accept()
Accepts dialog as though the user has pressed OK or similar button.

Cancel()
Cancels dialog as though the user has pressed Cancel.

Obtained From

UserInterface (ModalDialog)
UserInterfaceDialogSet (Item)

See Also

Scripting Reference
**UserInterfaceDialogSet Object**
Set of dialogs.

**Properties**

Application Application
Returns application. Read only.

Number Count
Returns number of dialogs in set. Read only.

UserInterfaceDialog Item(Number index)
UserInterfaceDialog Item(String name)
Returns dialog with given index or name. Read only.

Any Parent
Returns parent object. Read only.

**Methods**

Number ItemByName(String name)
Returns index of dialog with given name or -1.

**Obtained From**

UserInterface (Panels)

**See Also**

Scripting Reference
Version Object
Version information.

Properties

Application
Returns application. Read only.

Number Build
Returns build number. Read only.

String Edition
Returns edition string. Read only.

Number Major
Returns major version number. Read only.

Number Minor
Returns minor version number. Read only.

Boolean Prerelease
Returns True if application is a prerelease (beta) version and False otherwise. Read only.

Number ServicePack
Returns service pack number. Read only.

Any Parent
Returns parent object. Read only.

Obtained From

Application (Version)

See Also

Scripting Reference
View Object
Component view.

Properties

Application Application
Returns application. Read only.

Point Center
Sets or returns center of view in native coordinates. Read/write.

Component Component
Returns owner component. Read only.

String Name
Sets or returns view name. Read/write.

Any Parent
Returns parent object. Read only.

Number Scale
Sets or returns view scale with respect to current coordinate system. Read/write.

Number ScaleAbsolute
Sets or returns view scale. Read/write.

Obtained From

Application (NewView)
Drawing (DefaultView)
Image (DefaultView)
Labels (DefaultView)
Map (DefaultView)
Surface (DefaultView)
ViewSet (Item, LastAdded)
Zones (DefaultView)

See Also

Scripting Reference
ViewSet Object
Set of views.

Properties

Application Application
Returns application. Read only.

Number Count
Returns number of views in set. Read only.

View Item(Number index)
View Item(String name)
Returns view with given index or name. Read only.

View LastAdded
Returns last added view if any. Read only.

Any Parent
Returns parent object. Read only.

Methods

Add(View view)
Appends view to set.

Number ItemByName(String name)
Returns index of view with given name or -1.

Remove(Number index)
Removes view with given index.

RemoveAll()
Removes all views.

Obtained From

Drawing (ViewSet)
Image (ViewSet)
Labels (ViewSet)
Map (ViewSet)
Surface (ViewSet)
Zones (ViewSet)

See Also

Scripting Reference
**Window Object**

Generic window.

**Properties**

Component ActiveComponent
Returns active component shown in window. Read only.

Application Application
Returns application. Read only.

Rect Bounds
Returns the area covered by the window, in the coordinate system of the displayed component. Read only.

Component Component
Returns component shown in window. Read only.

Number Height
Returns window height in pixels. Read only.

Any Parent
Returns parent object. Read only.

Number Scale
Sets or returns viewport scale (projected components only). Read/write.

Number ScaleInternal
Sets or returns internal pixel-to-data ratio. Read/write.

Number Width
Returns window width in pixels. Read only.

**Methods**

Close()
Closes window.

Point GetLocation(Boolean latLon)
Returns viewport center. LatLon parameter is optional.

Boolean HasLocation()
Returns True if viewport center can be modified and False otherwise.

Boolean HasScale()
Returns True if viewport scale can be obtained or modified and False otherwise.

Boolean HasScaleInternal()
Returns True if internal pixel-to-data ratio can be obtained or modified and False otherwise.

MoveTo(Branch branch)
MoveTo(BranchSet branchSet)
MoveTo(Component component)
MoveTo(Geom geom)
MoveTo(Label label)
MoveTo(LabelSet labelSet)
MoveTo(Object object)
MoveTo(ObjectSet objectSet)
MoveTo(Pixel pixel)
MoveTo(PixelSet pixelSet)
MoveTo(Point point)
MoveTo(PointSet pointSet)
MoveTo(Rect rect)
Centers viewport on given object or set of objects. Branches, geoms, points, rects and their sets are assumed to be in the coordinate system of the displayed component.

MoveToLocation(Point point, Boolean latLon)
Centers viewport on given location. LatLon parameter is optional.

Refresh()
Updates window contents.

RenderTo(String name)
Renders window into an image with given name.

ZoomTo(Branch branch)
ZoomTo(BranchSet branchSet)
ZoomTo(Component component)
ZoomTo(Geom geom)
ZoomTo(Label label)
ZoomTo(LabelSet labelSet)
ZoomTo(Object object)
ZoomTo(ObjectSet objectSet)
ZoomTo(Pixel pixel)
ZoomTo(PixelSet pixelSet)
ZoomTo(Point point)
ZoomTo(PointSet pointSet)
ZoomTo(Rect rect)
Zooms to given object or set of objects. Branches, geoms, points, rects and their sets are assumed to be in the coordinate system of the displayed component. If the bounding box of the object is a point (for example, if the object is a point), centers viewport on given object.
WindowSet Object
Set of windows.

Properties

Window ActiveWindow
Returns active window. Read only.

Application Application
Returns application. Read only.

Number Count
Returns number of windows in set. Read only.

Window Item(Number index)
Returns window with given index. Read only.

Any Parent
Returns parent object. Read only.

Methods

Close()
Closes all windows in set.

Obtained From

Application (WindowSet)

See Also

Scripting Reference
Zone Object
Zone.

Properties

Application Application
Returns application. Read only.

Number Area
Returns zone area. Read only.

ZoneSet Children
Returns set of direct children of zone. Read only.

String Name
Returns name of zone. Read only.

Any Parent
Returns parent object. Read only.

Zone ParentZone
Returns parent zone. Read only

String Path
Returns path to zone. Read only.

Obtained From

Zone (ParentZone)
ZoneSet (Item)

See Also

Scripting Reference
**Zones Object**
Zones component.

**Properties**

**Application Application**
Returns application. Read only.

**ControlPointSet ControlPointSet**
Returns set of control points. Read only.

**CoordinateSystem CoordinateSystem**
Sets or returns coordinate system. Read/write.

**View DefaultView**
Returns default view. Read only.

**String Description**
Sets or returns component description. Read/write.

**Document Document**
Returns owner document. Read only.

**Folder Folder**
Sets or returns containing folder. Read/write.

**Number ID**
Returns unique component ID. Read only.

**String Name**
Sets or returns component name. Read/write.

**String Note**
Sets or returns notes associated with the component. Read/write.

**Component Owner**
Returns owner component if any. Read only.

**Any Parent**
Returns parent object. Read only.

**ZoneSelectionSet SavedSelectionSet**
Returns set of saved selections. Read only.

**ZoneSet Selection**
Returns set of selected zones. Read only.

**ComponentType Type**
Returns component type. Read only.

**String TypeName**
Returns textual representation of the component type. Read only.
ViewSet ViewSet
Returns set of views associated with this component. Read only.

Number ZoomMax
Sets or returns maximum zoom level. Read/write.

Number ZoomMin
Sets or returns minimum zoom level. Read/write.

ZoneSet ZoneSet
Returns set of zones. Read only.

**Methods**

Boolean IsEmpty()
Returns True if component contains no zones and False otherwise.

Open()
Opens component.

OpenInNewWindow()
Opens component in new window.

Print(Boolean useDialog)
Prints component.

**Obtained From**

Document (NewZones)

**See Also**

Scripting Reference
ZoneSelectionSet Object
Set of saved selections within zones.

Properties

Application Application
Returns application. Read only.

Number Count
Returns number of selections in set. Read only.

ZoneSet Item(Number index)
Returns selection with given index. Read only.

Any Parent
Returns parent object. Read only.

Methods

Number ItemByMask(Number mask)
Returns index of selection with given mask value or -1.

Number ItemByName(String name)
Returns index of selection with given name or -1.

Obtained From

Zones (SavedSelectionSet)

See Also

Scripting Reference
ZoneSet Object
Set of zones.

Properties

Application Application
Returns application. Read only.

Number Count
Returns number of zones in set. Read only.

Zone Item(Number index)
Returns zone with given index. Read only.

Any Parent
Returns parent object. Read only.

Obtained From

Zone (Children)
Zones (Selection, ZoneSet)
ZoneSelectionSet (Item)

See Also

Scripting Reference

Control Constants
ControlAlignment Constants
Control alignment options.

Values

wlLeft
Left alignment.

wlRight
Right alignment.

See Also

Scripting Reference
ControlAlignmentText Constants
Control text alignment options.

Values

wlCenter
Text is centered.

wlLeftJustify
Text is justified to the left side.

wlRightJustify
Text is justified to the right side.

See Also

Scripting Reference
ControlAppearance Constants

Control appearance options.

Values

wl3D
Control is displayed with 3D effect.

wlFlat
Control is displayed flat.

See Also

Scripting Reference
ControlBorderStyle Constants
Control border styles.

Values

wlFixedSingle
Control has single-line border.

wlNone
Control has no border.

See Also

Scripting Reference
ControlCheckValue Constants
Control check values.

Values

wlChecked
Control is checked.

wlGrayed
Control is grayed (neither checked nor unchecked).

wlUnchecked
Control is unchecked.

See Also

Scripting Reference
ControlMouseButton Constants
Mouse buttons recognized by component control.

Values

MouseButtonExtended1
First extended button.

MouseButtonExtended2
Second extended button.

MouseButtonLeft
Left button.

MouseButtonMiddle
Middle button.

MouseButtonNone
None.

MouseButtonRight
Right button.

See Also

Scripting Reference
ControlMouseMode Constants
Mouse modes supported by component control.

Values

ControlMouseModeNone
None.

ControlMouseModeCenter
Center.

ControlMouseModeGenericArea
Area. Uses BeginTrack and EndTrack events.

ControlMouseModeGenericAreaFreeform
Freeform area. Uses BeginTrack and EndTrack events.

ControlMouseModeGenericBox
Box. Uses BeginTrack and EndTrack events.

ControlMouseModeGenericBoxCenter
Box tracked from center. Uses BeginTrack and EndTrack events.

ControlMouseModeGenericCircle
Circle. Uses BeginTrack and EndTrack events.

ControlMouseModeGenericCircleBox
Circle tracked with a box. Uses BeginTrack and EndTrack events.

ControlMouseModeGenericCircleCenter
Circle tracked from center. Uses BeginTrack and EndTrack events.

ControlMouseModeGenericCircleDiameter
Circle tracked with a diameter. Uses BeginTrack and EndTrack events.

ControlMouseModeGenericLine
Line. Uses BeginTrack and EndTrack events.

ControlMouseModeGenericLineFreeform
Freeform line. Uses BeginTrack and EndTrack events.

ControlMouseModeGenericPoint
Point. Uses BeginTrack and EndTrack events.

ControlMouseModeGrabber
Grabber.

ControlMouseModeTracker
Tracker (measures distances and areas).

ControlMouseModeZoomBox
Zoom box.
ControlMouseModeZoomIn
Zoom in.

ControlMouseModeZoomOut
Zoom out.

See Also

Scripting Reference
ControlMousePointer Constants
Control mouse pointers.

Values

wlArrow
Arrow pointer.

wlArrowHourglass
Arrow with hourglass pointer.

wlArrowQuestion
Arrow with question pointer.

wlCross
Cross (precision selection) pointer.

wlCustom
Custom pointer.

wlDefault
Default pointer.

wlHourglass
Hourglass pointer.

wlBeam
I-beam (text editing) pointer.

wlIcon
Icon pointer.

wlNoDrop
“No-drop” pointer.

wlSize
Four arrow sizing pointer.

wlSizeAll
Four arrow sizing pointer.

wlSizeEW
Horizontal sizing pointer.

wlSizeNESW
Diagonal sizing pointer.

wlSizeNS
Vertical sizing pointer.

wlSizeNWSE
Diagonal sizing pointer.
wlUpArrow
Up arrow pointer.

See Also

Scripting Reference
**ControlMultiSelect Constants**
Control multiple selection options.

**Values**

- **wlExtended**
  Multiple selection is enabled. The user can select items by holding mouse button and dragging the cursor.

- **wlNoMultiSelect**
  Multiple selection is disabled.

- **wlSimple**
  Multiple selection is enabled.

**See Also**

Scripting Reference
ControlOleDrag Constants
Control OLE drag options.

Values

wlOLEDragAutomatic
Dragging is done in automatic mode.

wlOLEDragManual
Dragging is done in manual mode.

See Also

Scripting Reference
**ControlOleDrop Constants**
Control OLE drop options.

**Values**

- `wlOLEDropAutomatic`
  Dropping is done in automatic mode.

- `wlOLEDropManual`
  Dropping is done in manual mode.

- `wlOLEDropNone`
  Control does not support dropping.

**See Also**

Scripting Reference
ControlScrollBars Constants
Control scroll bar options.

Values

wlBoth
Control has both horizontal and vertical scroll bars.

wlHorizontal
Control has horizontal scroll bar.

wlNoScrollBars
Control has no scroll bars.

wlVertical
Control has vertical scroll bar.

See Also

Scripting Reference
**ControlStyle Constants**

Control styles.

**Values**

- **wlGraphical**
  Control is in graphical style and displays pictures.

- **wlStandard**
  Control is in standard style.

**See Also**

Scripting Reference
**ControlStyleCombo Constants**

Combo box control styles.

**Values**

wlDropdownCombo
Control has an edit portion and a list portion.

wlDropdownList
Control has an edit portion and a list portion. The edit portion can only display items from the list portion.

wlSimple
Control has an edit portion and a list portion that is always visible.

**See Also**

Scripting Reference
ControlStyleList Constants
List box control styles.

Values

wlCheckbox
Each control item has a checkbox near it.

wlStandard
Standard style.

See Also
Scripting Reference
Control Objects
Check Control
Check box control.

Properties

ControlAlignment Alignment
Sets or returns control alignment option. Read/write.

ControlAppearance Appearance
Sets or returns control appearance option. Read/write.

Number BackColor
Sets or returns background color. Read/write.

String Caption
Sets or returns control caption. Read/write.

Picture DisabledPicture
Sets or returns a graphic to be displayed when control is disabled. Read/write.

Picture DownPicture
Sets or returns a graphic to be displayed when control is pushed. Read/write.

Boolean Enabled
Sets or returns enabled option. Read/write.

Font Font
Sets or returns control font. Read/write.

Boolean FontBold
Sets or returns bold option of control font. Read/write.

Boolean FontItalic
Sets or returns italic option of control font. Read/write.

String FontName
Sets or returns facename of control font. Read/write.
Number FontSize
Sets or returns size of control font in points. Read/write.

Boolean FontStrikethru
Sets or returns strikethrough option of control font. Read/write.

Boolean FontUnderline
Sets or returns underline option of control font. Read/write.

Number ForeColor
Sets or returns foreground color. Read/write.

Number MaskColor
Sets or returns transparent color used in pictures. Read/write.

Picture MouseIcon
Sets or returns custom mouse icon. Read/write.

ControlMousePointer MousePointer
Sets or returns type of mouse pointer displayed over a control. Read/write.

ControlOleDrop OLEDropMode
Sets or returns OLE drop mode option. Read/write.

Picture Picture
Sets or returns a graphic to be displayed when control is up. Read/write.

Boolean RightToLeft
Sets or returns text display direction on a bidirectional system. Read/write.

ControlStyle Style
Sets or returns control style. Read/write.

Boolean UseMaskColor
Sets or returns transparent color option. Read/write.

ControlCheckValue Value
Sets or returns control value. Read/write.

**Methods**

OLEDrag()
Starts an OLE drag/drop event with given control as the source.

Refresh()
Forces a complete repaint of control.

**Events**

Click()
Occurs when the user clicks a control.
KeyDown(Number keyCode, Number shift)
Occurs when the user presses a key while a control has focus.

KeyPress(Number keyAscii)
Occurs when the user presses and releases an ANSI key.

KeyUp(Number keyCode, Number shift)
Occurs when the user releases a key while a control has focus.

MouseDown(Number button, Number shift, Number x, Number y)
Occurs when the user presses the mouse button over a control.

MouseMove(Number button, Number shift, Number x, Number y)
Occurs when the user moves the mouse.

MouseUp(Number button, Number shift, Number x, Number y)
Occurs when the user releases the mouse button.

See Also

Scripting Reference
Combo Control
Combo box control.

Properties

ControlAppearance Appearance
Sets or returns control appearance option. Read/write.

Number BackColor
Sets or returns background color. Read/write.

Boolean Enabled
Sets or returns enabled option. Read/write.

Font Font
Sets or returns control font. Read/write.

Boolean FontBold
Sets or returns bold option of control font. Read/write.

Boolean FontItalic
Sets or returns italic option of control font. Read/write.

String FontName
Sets or returns facename of control font. Read/write.

Number FontSize
Sets or returns size of control font in points. Read/write.

Boolean FontStrikethru
Sets or returns strikethrough option of control font. Read/write.

Boolean FontUnderline
Sets or returns underline option of control font. Read/write.

Number ForeColor
Sets or returns foreground color. Read/write.

Boolean IntegralHeight
Sets or returns integral height option. Value of false indicates that control displays partial items. Read/write.

Number ItemData(Number index)
Sets or returns a number associated with given control item. Read/write.

String List(Number index)
Sets or returns text of given control item. Read/write.

Number ListCount
Sets or returns number of items in control. Read/write.

Number ListIndex
Sets or returns index of currently selected control item. Read/write.
Boolean Locked
Sets or returns control locking option. Locked controls cannot be edited. Read/write.

Picture MouseIcon
Sets or returns custom mouse icon. Read/write.

ControlMousePointer MousePointer
Sets or returns type of mouse pointer displayed over a control. Read/write.

Number NewIndex
Sets or returns index of most recently added control item. Read/write.

ControlOleDrag OLEDragMode
Sets or returns OLE drag mode option. Read/write.

ControlOleDrop OLEDropMode
Sets or returns OLE drop mode option. Read/write.

Boolean RightToLeft
Sets or returns text display direction on a bidirectional system. Read/write.

Number SelLength
Sets or returns number of characters in text selection. Read/write.

Number SelStart
Sets or returns starting point of text selection. Read/write.

String SelText
Sets or returns text selection. Read/write.

Boolean Sorted
Sets or returns sorted control option. Read/write.

ControlStyleCombo Style
Sets or returns control style. Read/write.

String Text
Sets or returns the text contained in control. Read/write.

Number TopIndex
Sets or returns index of topmost control item. Read/write.

Methods

Number AddItem(String item)
Appends new control item and returns its index.

Clear()
Removes all control items.

OLEDrag()
Starts an OLE drag/drop event with given control as the source.
Refresh()
Forces a complete repaint of control.

RemoveItem(Number index)
Removes item with given index.

Events

Change()
Occurs when the contents of a control has changed.

Click()
Occurs when the user clicks a control.

DbClick()
Occurs when the user doubleclicks a control.

DropDown()
Occurs when the list portion of a control is about to drop down.

KeyDown(Number keyCode, Number shift)
Occurs when the user presses a key while a control has focus.

KeyPress(Number keyAscii)
Occurs when the user presses and releases an ANSI key.

KeyUp(Number keyCode, Number shift)
Occurs when the user releases a key while a control has focus.

Scroll(Number keyCode, Number shift)
Occurs when the user scrolls the contents of a control.

See Also

Scripting Reference
**Command Control**
Command button control.

**Properties**

**ControlAppearance Appearance**
Sets or returns control appearance option. Read/write.

**Number BackColor**
Sets or returns background color. Read/write.

**String Caption**
Sets or returns control caption. Read/write.

**Picture DisabledPicture**
Sets or returns a graphic to be displayed when control is disabled. Read/write.

**Picture DownPicture**
Sets or returns a graphic to be displayed when control is pushed. Read/write.

**Boolean Enabled**
Sets or returns enabled option. Read/write.

**Font Font**
Sets or returns control font. Read/write.

**Boolean FontBold**
Sets or returns bold option of control font. Read/write.

**Boolean FontItalic**
Sets or returns italic option of control font. Read/write.

**String FontName**
Sets or returns facename of control font. Read/write.

**Number FontSize**
Sets or returns size of control font in points. Read/write.

**Boolean FontStrikethru**
Sets or returns strikethrough option of control font. Read/write.

**Boolean FontUnderline**
Sets or returns underline option of control font. Read/write.

**Number ForeColor**
Sets or returns foreground color. Read/write.

**Number MaskColor**
Sets or returns transparent color used in pictures. Read/write.

**Picture MouseIcon**
Sets or returns custom mouse icon. Read/write.
ControlMousePointer MousePointer
Sets or returns type of mouse pointer displayed over a control. Read/write.

ControlOleDrop OLEDropMode
Sets or returns OLE drop mode option. Read/write.

Picture Picture
Sets or returns a graphic to be displayed when control is up. Read/write.

Boolean RightToLeft
Sets or returns text display direction on a bidirectional system. Read/write.

ControlStyle Style
Sets or returns control style. Read/write.

Boolean UseMaskColor
Sets or returns transparent color option. Read/write.

Boolean Value
Sets or returns control value. Read/write.

Methods

DoClick()
Triggers the click event.

OLEDrag()
Starts an OLE drag/drop event with given control as the source.

Refresh()
Forces a complete repaint of control.

Events

Click()
Occurs when the user clicks a control.

KeyDown(Number keyCode, Number shift)
Occurs when the user presses a key while a control has focus.

KeyPress(Number keyAscii)
Occurs when the user presses and releases an ANSI key.

KeyUp(Number keyCode, Number shift)
Occurs when the user releases a key while a control has focus.

MouseDown(Number button, Number shift, Number x, Number y)
Occurs when the user presses the mouse button over a control.

MouseMove(Number button, Number shift, Number x, Number y)
Occurs when the user moves the mouse.

MouseUp(Number button, Number shift, Number x, Number y)
Occurs when the user releases the mouse button.
See Also

Scripting Reference
Component Control
Component control. Shows drawings, images, labels, maps, surfaces, and other components.

Properties

Point Center
Sets or returns center of view. Read/write.

Component Component
Sets or returns displayed component. Read/write.

String ComponentName
Sets or returns name of displayed component. Read/write.

Rect DataBounds
Sets or returns view bounds. Read/write.

Document Document
Sets or returns displayed document. Read/write.

String DocumentPath
Sets or returns path to displayed document. Read/write.

ControlMouseMode MouseMode
Sets or returns mouse mode. Read/write.

Boolean RenderBackground
Sets or returns background rendering mode. Read/write.

Boolean RenderBorder
Sets or returns border rendering mode. Read/write.

Boolean RenderControlPoints
Sets or returns control points rendering mode. Read/write.

Boolean RenderGraticule
Sets or returns graticule rendering mode. Read/write.

Boolean RenderGrid
Sets or returns grid rendering mode. Read/write.

Boolean RenderLegend
Sets or returns legend rendering mode. Read/write.

Boolean RenderNorthArrow
Sets or returns north arrow rendering mode. Read/write.

Boolean RenderScaleBar
Sets or returns scale bar rendering mode. Read/write.

Boolean RenderSelection
Sets or returns selection rendering mode. Read/write.
Number Scale
Sets or returns view scale. Read/write.

Number ScaleAbsolute
Sets or returns absolute view scale (e.g., 1:1000) for projected components. Read/write.

Boolean StatusBar
Shows or hides status bar. Read/write.

Boolean ToolBar
Shows or hides toolbar. Read/write.

**Methods**

Label GetLabelAt(Point point)
Retrieves label at given screen coordinates.

Object GetObjectAt(Point point)
Retrieves object at given screen coordinates.

GoTo(Branch branch)
GoTo(BranchSet branchSet)
GoTo(Component component)
GoTo(Geom geom)
GoTo(Label label)
GoTo(LabelSet labelSet)
GoTo(Object object)
GoTo(ObjectSet objectSet)
GoTo(Pixel pixel)
GoTo(PixelSet pixelSet)
GoTo(Point point)
GoTo(Rect rect)
Centers view on given branch, branch set, component, geom, label, label set, object, object set, pixel, pixel set, point, or rect.

Boolean HasLabelAt(Point point)
Check if there is a label at given screen coordinates.

Boolean HasObjectAt(Point point)
Check if there is an object at given screen coordinates.

Point NativeToScreen(Point point)
Converts point from the coordinate system of the displayed component to screen coordinates.

Refresh()
Repaints the control.

Point ScreenToNative(Point point)
Converts point from screen coordinates to the coordinate system of the displayed component.

ZoomIn(Number factor, Point center)
Zooms in with a given factor and center. Both parameters are optional.
ZoomOut(Number factor, Point center)
Zooms out with a given factor and center. Both parameters are optional.

ZoomTo(Branch branch)
ZoomTo(BranchSet branchSet)
ZoomTo(Component component)
ZoomTo(Geom geom)
ZoomTo(Label label)
ZoomTo(LabelSet labelSet)
ZoomTo(Object object)
ZoomTo(ObjectSet objectSet)
ZoomTo(PixelSet pixelSet)
ZoomTo(Rect rect)
Zooms to given branch, branch set, component, geom, label, label set, object, object set, pixel set, or rect.

ZoomToFit()
Zooms to fit entire component.

Events

BeginTrack(ControlMouseEventArgs args)
Occurs when the user starts tracking shape.

Click(ControlMouseEventArgs args)
Occurs when the user clicks a control.

DoubleClick(ControlMouseEventArgs args)
Occurs when the user double clicks a control.

EndTrack(ControlTracksEventArgs args)
Occurs when the user stops tracking shape.

MouseDown(ControlMouseEventArgs args)
Occurs when the user presses the mouse button within a control.

MouseMove(ControlMouseEventArgs args)
Occurs when the user moves the mouse within a control.

MouseUp(ControlMouseEventArgs args)
Occurs when the user releases the mouse button within a control.

MouseWheel(ControlMouseEventArgs args)
Occurs when the user scrolls the mouse wheel within a control.

See Also

Scripting Reference
**ControlMouseEventArgs Object**
Arguments of a mouse event raised by component control.

**Properties**

**ControlMouseButton Button**
Returns pressed or released mouse button. Read only.

**Boolean Canceled**
Accepts or cancels event. Read/write.

**Point LocationLatLon**
Returns event location in lat/lon coordinates. Read only.

**Point LocationNative**
Returns event location in component coordinates. Read only.

**Point LocationScreen**
Returns event location in screen coordinates. Read only.

**String Report**
Returns event report.

**Number WheelDelta**
The distance the wheel is rotated. A positive value indicates that the wheel was rotated forward, away from the user; a negative value indicates that the wheel was rotated backward, toward the user. Read only.

**See Also**

**ComponentControl** (BeginTrack, Click, DoubleClick, MouseDown, MouseMove, MouseUp)

Scripting Reference
ControlTrackEventArgs Object
Arguments of a mouse event raised by component control.

Properties

Geom GeomLatLon
Returns tracked shape in lat/lon coordinates. Read only.

Geom GeomNative
Returns tracked shape in component coordinates. Read only.

Geom GeomScreen
Returns tracked shape in screen coordinates. Read only.

See Also

ComponentControl (EndTrack)
Scripting Reference
Frame Control
Frame control.

Properties

ControlAppearance Appearance
Sets or returns control appearance option. Read/write.

Number BackColor
Sets or returns background color. Read/write.

ControlBorderStyle BorderStyle
Sets or returns control border style. Read/write.

String Caption
Sets or returns control caption. Read/write.

Boolean ClipControls
Sets or returns clipping option. When clipping is set to True, the frame clips contained controls when painting. Read/write.

Boolean Enabled
Sets or returns enabled option. Read/write.

Font Font
Sets or returns control font. Read/write.

Boolean FontBold
Sets or returns bold option of control font. Read/write.

Boolean FontItalic
Sets or returns italic option of control font. Read/write.

String FontName
Sets or returns facename of control font. Read/write.

Number FontSize
Sets or returns size of control font in points. Read/write.

Boolean FontStrikethru
Sets or returns strikethrough option of control font. Read/write.

Boolean FontUnderline
Sets or returns underline option of control font. Read/write.

Number ForeColor
Sets or returns foreground color. Read/write.

Picture MouseIcon
Sets or returns custom mouse icon. Read/write.

ControlMousePointer MousePointer
Sets or returns type of mouse pointer displayed over a control. Read/write.
ControlOleDrop OLEDropMode
Sets or returns OLE drop mode option. Read/write.

Boolean RightToLeft
Sets or returns text display direction on a bidirectional system. Read/write.

Methods

OLEDrag()
Starts an OLE drag/drop event with given control as the source.

Refresh()
Forces a complete repaint of control.

Events

Click()
Occurs when the user clicks a control.

DbClick()
Occurs when the user doubleclicks a control.

MouseDown(Number button, Number shift, Number x, Number y)
Occurs when the user presses the mouse button over a control.

MouseMove(Number button, Number shift, Number x, Number y)
Occurs when the user moves the mouse.

MouseUp(Number button, Number shift, Number x, Number y)
Occurs when the user releases the mouse button.

See Also

Scripting Reference
HScroll Control
Horizontal scrollbar control.

Properties

Boolean Enabled
Sets or returns enabled option. Read/write.

Number LargeChange
Sets or returns the amount of change to value when the user clicks the scroll bar area. Read/write.

Number Max
Sets or returns maximum value. Read/write.

Number Min
Sets or returns minimum value. Read/write.

Picture MouseIcon
Sets or returns custom mouse icon. Read/write.

ControlMousePointer MousePointer
Sets or returns type of mouse pointer displayed over an control. Read/write.

Boolean RightToLeft
Sets or returns text display direction on a bidirectional system. Read/write.

Number SmallChange
Sets or returns the amount of change to value when the user clicks the scroll bar arrow. Read/write.

Number Value
Sets or returns current value. Read/write.

Methods

Refresh()
Forces a complete repaint of control.

Events

Change()
Occurs when the contents of a control has changed.

KeyDown(Number keyCode, Number shift)
Occurs when the user presses a key while a control has focus.

KeyPress(Number keyAscii)
Occurs when the user presses and releases an ANSI key.

KeyUp(Number keyCode, Number shift)
Occurs when the user releases a key while a control has focus.

Scroll()
Occurs when the user scrolls a control.

See Also

Scripting Reference
**List Control**
List box control.

**Properties**

*ControlAppearance Appearance*
Sets or returns control appearance option. Read/write.

*Number BackColor*
Sets or returns background color. Read/write.

*Number Columns*
Sets or returns columns option. Value of 0 means that the control is scrolled vertically, values greater than 0 mean that the control is scrolled horizontally in snaking columns. Read/write.

*Boolean Enabled*
Sets or returns enabled option. Read/write.

*Font Font*
Sets or returns control font. Read/write.

*Boolean FontBold*
Sets or returns bold option of control font. Read/write.

*Boolean FontItalic*
Sets or returns italic option of control font. Read/write.

*String FontName*
Sets or returns facename of control font. Read/write.

*Number FontSize*
Sets or returns size of control font in points. Read/write.

*Boolean FontStrikethru*
Sets or returns strikethrough option of control font. Read/write.

*Boolean FontUnderline*
Sets or returns underline option of control font. Read/write.

*Number ForeColor*
Sets or returns foreground color. Read/write.

*Boolean IntegralHeight*
Sets or returns integral height option. Values of false indicated that control displays partial items. Read/write.

*Number ItemData(Number index)*
Sets or returns a number associated with given control item. Read/write.

*String List(Number index)*
Sets or returns text of given control item. Read/write.

*Number ListCount*
Sets or returns number of items in control. Read/write.
Number ListIndex
Sets or returns index of currently selected control item. Read/write.

Picture MouseIcon
Sets or returns custom mouse icon. Read/write.

ControlMousePointer MousePointer
Sets or returns type of mouse pointer displayed over an control. Read/write.

ControlMultiSelect MultiSelect
Sets or returns selection option. Read/write.

Number NewIndex
Sets or returns index of most recently added control item. Read/write.

ControlOleDrag OLEDragMode
Sets or returns OLE drag mode option. Read/write.

ControlOleDrop OLEDropMode
Sets or returns OLE drop mode option. Read/write.

Boolean RightToLeft
Sets or returns text display direction on a bidirectional system. Read/write.

Boolean Selected(Number item)
Sets or returns selection state of item with given index. Read/write.

Number SelCount
Sets or returns number of selected items in control. Read/write.

Boolean Sorted
Sets or returns sorted control option. Read/write.

ControlStyleList Style
Sets or returns control style. Read/write.

String Text
Sets or returns the text contained in control. Read/write.

Number TopIndex
Sets or returns index of topmost control item. Read/write.

Methods

Number AddItem(String item)
Appends new control item and returns its index.

Clear()
Removes all control items.

OLEDrag()
Starts an OLE drag/drop event with given control as the source.
Refresh()
Forces a complete repaint of control.

RemoveItem(Number index)
Removes item with given index.

Events

Click()
Occurs when the user clicks a control.

DbClick()
Occurs when the user doubleclicks a control.

ItemCheck()
Occurs when the user checks or unchecks control item (check list box controls only).

KeyDown(Number keyCode, Number shift)
Occurs when the user presses a key while a control has focus.

KeyPress(Number keyAscii)
Occurs when the user presses and releases an ANSI key.

KeyUp(Number keyCode, Number shift)
Occurs when the user releases a key while a control has focus.

MouseDown(Number button, Number shift, Number x, Number y)
Occurs when the user presses the mouse button over a control.

MouseMove(Number button, Number shift, Number x, Number y)
Occurs when the user moves the mouse.

MouseUp(Number button, Number shift, Number x, Number y)
Occurs when the user releases the mouse button.

Scroll(Number keyCode, Number shift)
Occurs when the user scrolls the contents of a control.

See Also

Scripting Reference
**Option Control**

Option button control.

**Properties**

ControlAlignment Alignment
Sets or returns control alignment option. Read/write.

ControlAppearance Appearance
Sets or returns control appearance option. Read/write.

Number BackColor
Sets or returns background color. Read/write.

String Caption
Sets or returns control caption. Read/write.

Picture DisabledPicture
Sets or returns a graphic to be displayed when control is disabled. Read/write.

Picture DownPicture
Sets or returns a graphic to be displayed when control is pushed. Read/write.

Boolean Enabled
Sets or returns enabled option. Read/write.

Font Font
Sets or returns control font. Read/write.

Boolean FontBold
Sets or returns bold option of control font. Read/write.

Boolean FontItalic
Sets or returns italic option of control font. Read/write.

String FontName
Sets or returns facename of control font. Read/write.

Number FontSize
Sets or returns size of control font in points. Read/write.

Boolean FontStrikethru
Sets or returns strikethrough option of control font. Read/write.

Boolean FontUnderline
Sets or returns underline option of control font. Read/write.

Number ForeColor
Sets or returns foreground color. Read/write.

String Group
Sets or returns control group. Read/write.
Number MaskColor
Sets or returns transparent color used in pictures. Read/write.

Picture MouseIcon
Sets or returns custom mouse icon. Read/write.

ControlMousePointer MousePointer
Sets or returns type of mouse pointer displayed over a control. Read/write.

ControlOleDrop OLEDropMode
Sets or returns OLE drop mode option. Read/write.

Picture Picture
Sets or returns a graphic to be displayed when control is up. Read/write.

Boolean RightToLeft
Sets or returns text display direction on a bidirectional system. Read/write.

ControlStyle Style
Sets or returns control style. Read/write.

Boolean UseMaskColor
Sets or returns transparent color option. Read/write.

Boolean Value
Sets or returns control value. Read/write.

Methods

OLEDrag()
Starts an OLE drag/drop event with given control as the source.

Refresh()
Forces a complete repaint of control.

Events

Click()
Occurs when the user clicks a control.

DblClick()
Occurs when the user doubleclicks a control.

KeyDown(Number keyCode, Number shift)
Occurs when the user presses a key while a control has focus.

KeyPress(Number keyAscii)
Occurs when the user presses and releases an ANSI key.

KeyUp(Number keyCode, Number shift)
Occurs when the user releases a key while a control has focus.

MouseDown(Number button, Number shift, Number x, Number y)
Occurs when the user presses the mouse button over a control.

MouseMove(Number button, Number shift, Number x, Number y)
Occurs when the user moves the mouse.

MouseUp(Number button, Number shift, Number x, Number y)
Occurs when the user releases the mouse button.

See Also

Scripting Reference
**Text Control**

Text box control.

**Properties**

- **ControlAlignment** [Text Alignment]
  Sets or returns text alignment option. Read/write.

- **ControlAppearance** [Appearance]
  Sets or returns control appearance option. Read/write.

- **Number BackColor**
  Sets or returns background color. Read/write.

- **ControlBorderStyle** [BorderStyle]
  Sets or returns control border style. Read/write.

- **Boolean Enabled**
  Sets or returns enabled option. Read/write.

- **Font Font**
  Sets or returns control font. Read/write.

- **Boolean FontBold**
  Sets or returns bold option of control font. Read/write.

- **Boolean FontItalic**
  Sets or returns italic option of control font. Read/write.

- **String FontName**
  Sets or returns facename of control font. Read/write.

- **Number FontSize**
  Sets or returns size of control font in points. Read/write.

- **Boolean FontStrikethru**
  Sets or returns strikethrough option of control font. Read/write.

- **Boolean FontUnderline**
  Sets or returns underline option of control font. Read/write.

- **Number ForeColor**
  Sets or returns foreground color. Read/write.

- **Boolean HideSelection**
  Sets or returns selection hiding option. When set to True, the control hides selection when it does not have focus. Read/write.

- **Boolean Locked**
  Sets or returns control locking option. Locked controls can not be edited. Read/write.

- **Number MaxLength**
  Sets or returns maximum number of characters that can be entered in a control. Read/write.
Picture MouseIcon
Sets or returns custom mouse icon. Read/write.

ControlMousePointer MousePointer
Sets or returns type of mouse pointer displayed over an control. Read/write.

Boolean MultiLine
Sets or returns control multiline option. Read/write.

ControlOleDrag OLEDragMode
Sets or returns OLE drag mode option. Read/write.

ControlOleDrop OLEDropMode
Sets or returns OLE drop mode option. Read/write.

String PasswordChar
Sets or returns password character. Read/write.

Boolean RightToLeft
Sets or returns text display direction on a bidirectional system. Read/write.

ControlScrollBars ScrollBars
Sets or returns control scroll bar options. Read/write.

Number SelLength
Sets or returns number of characters in text selection. Read/write.

Number SelStart
Sets or returns starting point of text selection. Read/write.

String SelText
Sets or returns text selection. Read/write.

String Text
Sets or returns the text contained in control. Read/write.

Methods

OLEDrag()
Starts an OLE drag/drop event with given control as the source.

Refresh()
Forces a complete repaint of control.

Events

Change()
Occurs when the contents of a control has changed.

Click()
Occurs when the user clicks a control.
DblClick()  
Occurs when the user double-clicks a control.

KeyDown(Number keyCode, Number shift)  
Occurs when the user presses a key while a control has focus.

KeyPress(Number keyAscii)  
Occurs when the user presses and releases an ANSI key.

KeyUp(Number keyCode, Number shift)  
Occurs when the user releases a key while a control has focus.

MouseDown(Number button, Number shift, Number x, Number y)  
Occurs when the user presses the mouse button over a control.

MouseMove(Number button, Number shift, Number x, Number y)  
Occurs when the user moves the mouse.

MouseUp(Number button, Number shift, Number x, Number y)  
Occurs when the user releases the mouse button.

See Also

Scripting Reference
VScroll Control
Vertical scrollbar control.

Properties

Boolean Enabled
Sets or returns enabled option. Read/write.

Number LargeChange
Sets or returns the amount of change to value when the user clicks the scroll bar area. Read/write.

Number Max
Sets or returns maximum value. Read/write.

Number Min
Sets or returns minimum value. Read/write.

Picture MouseIcon
Sets or returns custom mouse icon. Read/write.

ControlMousePointer MousePointer
Sets or returns type of mouse pointer displayed over an control. Read/write.

Boolean RightToLeft
Sets or returns text display direction on a bidirectional system. Read/write.

Number SmallChange
Sets or returns the amount of change to value when the user clicks the scroll bar arrow. Read/write.

Number Value
Sets or returns current value. Read/write.

Methods

Refresh()
Forces a complete repaint of control.

Events

Change()
Occurs when the contents of a control has changed.

KeyDown(Number keyCode, Number shift)
Occurs when the user presses a key while a control has focus.

KeyPress(Number keyAscii)
Occurs when the user presses and releases an ANSI key.

KeyUp(Number keyCode, Number shift)
Occurs when the user releases a key while a control has focus.

Scroll()
Occurs when the user scrolls a control.

**See Also**

Scripting Reference
Menus, Dialogs and Controls Reference

The topics in this section provide a reference to main menu choices, dialogs, toolbars, panes and context menus that occur with Manifold System. Read these topics together with overview topics found in the Introduction and general topics for each component type, such as Drawings, Images and so on.

To read all of the topics in the Menus, Dialogs and Controls chapters in Help, use the browse buttons to read them in sequence. Alternatively, press the Help Topics button in the Help system toolbar and in the Contents tab expand the Menus, Dialogs and Controls Reference headings to find the topic you would like to read.

File Menu

File - New / Open / Close

These File commands refer to Manifold projects. Individual components such as drawings, images and maps are imported or exported or created via the File - Export, File - Import and File - Link commands or the Create button in the project pane.

- **New** Create a new project.
- **Open** Open an existing project.
- **Close** Close the current project. Manifold will ask if we want to save the project if it has not been saved since the last change made.

Top open an existing project, browse the Windows file system to locate the Manifold project you wish to open. Click on a file to load it into the File name box and push Open, or simply double-click on the desired file.

- **Open as read-only** Check this box to open the specified project file "read only." This will allow viewing but will prevent any changes to the file.
**File - Save / Save As**

Save the current project under the given name. Browse the Windows file system to locate the folder in which you wish to save the project. Enter the desired name without a `.map` extension in the **File name** box and press **Enter** or push the **Save** button. The project will be saved to that name.

The project will now be known by that name. Any future Save commands will save the project to this file.

By default, Manifold will compress `.map` files to save space on disk. When working with large files the time required to decompress the `.map` file could be significant. For faster loads and saves, open the Tools - Options - **Miscellaneous** dialog and uncheck the **Compress .map files to save space** checkbox.
**File - Create**

The **File - Create** menu allows us to create new components in the project, for example, new blank drawings or images. It is frequently used to create new maps.

Some components that may be created must be created based upon some other component. For example, a new map must always be based on at least one existing drawing or image. The new map takes its initial projection properties from the component used to create it. These may later be changed if desired.

**Very Important:** To facilitate the creation of new components with projections that are a good match to components already in use, the **File - Create** commands will use the projection of the active window as the default projection for newly created components:

- If the project pane is the active window when **File - Create** commands are invoked, then the new drawings, images and text components will be created with a default projection of Orthographic centered at the 0,0 origin of the world latitude and longitude system.
- If some other window (such as a map window) is active, the **File - Create** commands will use that window's projection properties for the new component.
File - Export

Save the current component (the one opened in the active window) under the given name. A copy of the data for the component will be saved in the desired format. This command is most often used to export drawings, images and tables.

Browse the Windows file system to locate the folder in which you wish to save the component. Choose the desired file type from the list available in the Files of type box. Enter the desired name in the File name box without an extension and press Enter or push the Save button. The component will be saved to that name using the correct extension for that file type.

Files of type A pull-down box with a list of file types to which Manifold can export for this type of component. Choose the file type you wish to create.

The Files of type box will be loaded by default with the last type of file format that was exported.

When using the File - Export - Drawing dialog the Files of type choice acts as a filter for what is displayed in the browse pane in the Export dialog. Simple file types are exported directly from the Export dialog by choosing the desired type in the Files of type box. More complex exports into data sources such as databases are exported by choosing Data Sources () in the Files of type box and then using the Data Source dialog.

Drawings, images and surfaces can be exported into databases using spatial DBMS technology by using the Data Source dialog. See the Export Drawing - Oracle topic for a typical example of exporting to a spatial DBMS as can also be done for DB 2 with Spatial Extender, PostgreSQL/PostGIS, SQL Server 2008, SQL Server 2005 with Manifold Spatial Extender, and just about any DBMS using Manifold's ability to create and use generic spatial indices.

Using Enterprise Edition, any component can be exported into an Enterprise Server for shared storage.

Notes on Saving Components

Once components (like drawings and images) are imported into Manifold they become part of the project .map file. Saving the project saves the components within the .map file. At times we would like to import images, drawings or tables into Manifold and to later save them as independent drawing, image or table files.

To do this, we open the component in a window and then use the File - Export command to export it to the desired format. For example, we may import drawings in .shp format, manipulate them within Manifold, and then use File - Export to save them in .shp format for use with other software.

Note that not all formats can handle the full range of information possible in Manifold components. For example, tables in Manifold can contain field names that are fine for export to Access .mdb files but which are longer than allowed in the .dbf format used in dBase files. Such field names should be changed before attempting export to .dbf format.

Since .shp "shapefiles" are actually an ensemble of files that include the use of .dbf files for data attributes, this means that exporting a drawing (and, of course, its associated table) to .shp format means that field names in tables may need to be shortened. See the Export Drawing - SHP, Shapefiles topic.

By default, Manifold includes many more formats available for import than for export. Although all Manifold editions read .e00 format, only Enterprise Edition and higher can write .e00. Exporting to spatial DBMS usually requires Enterprise Edition.

Very Important

Before using Export with an image linked from an image server, see the Very Important discussion in the Image - Unlink command.

See Also
Import and Export …and subsequent topics in the Import and Export chapter in Help.

Exporting Layouts

The Data Source Dialog
Enterprise Edition
Spatial DBMS
**File - Import**

Find, open and import a component file into the project. Most components imported are drawings, images, tables or terrains.

Choose the desired file type from the list available in the **Files of type** box. Only those file types that may be used for the component being imported will be displayed. Browse the Windows file system to locate the file you wish to open. Click on a file to load it into the File name box and push **Open**, or simply double-click on the desired file. Importing a component brings a copy of the component's data into the project and will increase the size of the project file.

**Files of type**  
A pull-down box with a list of importable file types. Choose the file type you are searching for. Only those file types will be visible in the file-browsing window.

The **Files of type** box will be loaded by default with the last type of drawing file that was imported.

When using the **File - Export - Drawing** dialog the **Files of type** choice acts as a filter for what is displayed in the browse pane in the **Export** dialog. Simple file types are imported directly from the **Export** dialog by choosing the desired type in the **Files of type** box. More complex exports into data sources such as databases may be imported by choosing **Data Sources ()** in the **Files of type** box and then using the Data Source dialog.

Drawings, images and surfaces can be imported from databases using spatial DBMS technology by using the Database Console.

See the example topics referenced by the Manifold Spatial DBMS Facilities topic for typical examples of exporting and importing using a spatial DBMS as can be done with Oracle, DB 2 with Spatial Extender, PostgreSQL/PostGIS, SQL Server 2008, SQL Server 2005 with Manifold Spatial Extender, and just about any DBMS using Manifold's ability to create and use generic spatial indices.

Using Enterprise Edition, any component can be imported from an **Enterprise Server** providing shared storage.

**See Also**

**File - Import - Component**  
**File - Import - Component from Server**

Import and Export …and subsequent topics in the **Import and Export** chapter in Help.

**Exporting Layouts**

**The Data Source Dialog**

**Enterprise Edition**  
**Spatial DBMS**
File - Import - Component

Find, open and import a component file into the project from an existing Manifold .map project file. In the Import dialog navigate to a folder containing .map files and choose one of them in the File name box. Press Open. The map file will be decompressed and examined to see what components it contains and a list of these will be displayed in the Import MAP File dialog.

Check the components to be imported and press OK.

Import MAP File Dialog Commands

- Select All - Check all components for import.
- Select None - Uncheck all components.
- Select Inverse - Uncheck all checked components and check all unchecked components. A fast way to use all but one component: click Select None, check the one component not desired and then click Select Inverse.

See Also

File - Import - Component from Server
File - Import - Component from Server

Available in Manifold System Enterprise Edition only. If you have not licensed Enterprise Edition this command will not be available.

Find, open and import a component file into the project from an existing Manifold Enterprise server. Specify an Enterprise server in the **Server** box by entering the name of the data source or by entering the connection string, or browse to an ODBC data source by pressing the browse button next to the **Server** box. The box will remember the last-used Enterprise server connection.

Check the components to be imported and press **OK**. Checking or unchecking a component will check or uncheck all subsidiary components (such as labels components created using drawings and their tables). Hold down the **CTRL** key when checking or unchecking to check/uncheck a component without affecting the subsidiary components.

**Import from Server Dialog Commands**

- **Refresh** - Update the components pane with the latest contents of the Enterprise server. Use if the dialog has been open for a while and another user may have made changes from a different system.

- **Select All** - Check all components for import.

- **Select None** - Uncheck all components.

- **Select Inverse** - Uncheck all checked components and check all unchecked components. A fast way to use all but one component: click **Select None**, check the one component not desired and then click **Select Inverse**.
File - Link

Use the File - Link - Table dialog to link tables, drawings, images or surfaces into a Manifold project dialog. In addition, tables, drawings or images stored within databases using spatial DBMS technology can be linked by connecting to the database using Database Console, clicking on the table to highlight it and then clicking the Link button in the Database Console toolbar.

When using the File - Link - Table dialog the Files of type choice acts as a filter for what is displayed in the browse pane in the Link dialog. Simple file types are linked directly from the Link dialog by choosing the desired type in the Files of type box. More complex links from data sources such as databases are linked by choosing Data Sources () in the Files of type box and then using the Data Source dialog.

See the Importing and Linking Tables for information on linked tables.

Linked Drawings

A linked drawing is a special kind of drawing that is created on the fly from the information in a table or query. The table or query may be in the Manifold project or it may be a table or query in an external database provider. Linked drawings are shown in the project pane using an icon that include a yellow "database" cylinder to show they are created from a database table or query. Linked drawings can show points or lines.

All linked drawings can be formatted. In addition, in many cases a linked drawing can store formatting on the data source it has been linked from. This functionality has to be explicitly enabled by a database administrator using Database Administrator Edition. Linked drawings can be used to create subsidiary components, such as themes or labels.

There are two types of linked drawings in Manifold. One type of linked drawings is created directly from geocoded tables. An even more powerful form of linked drawings is the use of linked drawings created from tables that store geometry data. Linked drawings may be read-only or read/write depending upon their source.

Linked drawings that are linked from geometry in tables are normally read/write and may be fully edited just like a drawing in the local projection, including concurrent, multi-user editing of the linked drawing.

Linked drawings that are linked from geocoded tables are read-only and cannot be edited to move points or lines or to add or delete points or lines, except by editing the external table that controls them.

See these topics for information on linked drawings:

Linked Drawings
Geometry in Tables
Queries and Geoms

Linked Surfaces

Linked surfaces are like linked drawings in that they are created on the fly from the information in a table or query. The table or query may be in the Manifold project or it may be a table or query in an external database provider. Linked surfaces are shown in the project pane using an icon that include a yellow "database" cylinder to show they are created from a database table or query. Linked surfaces are normally read-only.

When a surface is linked from a table or query, each record in the table or query produces a pixel. The coordinates of the pixel are taken from the X and Y columns (cast to integer) specified in the Link dialog. The height of the pixel is taken from the Height column.

See these topics for information on linked surfaces:

Linked Surfaces
Virtual Tables for Images and Surfaces
Queries and Images or Surfaces
Geometry in Tables
Queries and Geoms
Linked Images

Linked images may be linked from several different sources. They may be created on the fly from information in a table or query, which may be inside the Manifold project or taken from an external database provider. In addition, linked images may be created by linking to a compressed image or by connecting to an image server that serves compressed images. Linked images are normally read-only. Finally, an image may be linked from an Oracle spatial data source that supports GeoRaster storage.

When an image is linked from a table or query, each record in the table or query produces a pixel. The coordinates of the pixel are taken from the X and Y columns (cast to integer) specified in the Link dialog. The color of the pixel is taken from either the Color column or the channel columns.

See these topics for information on linked images:

Compressed Images
Linked Images
Virtual Tables for Images and Surfaces
Queries and Images or Surfaces
Geometry in Tables
Queries and Geoms
File - Page Setup

The File - Page Setup command allows us to set paper orientation (portrait or landscape), paper size, printer, printer properties and margins. Page setup options will be applied to any print job and will be automatically incorporated into Layouts that may be created.

The Page Setup dialog uses inches or millimeters depending on the setting of the Use English Measurement Units checkbox in the Tools - Options dialog.

Note

Some multipage layouts of large maps using white backgrounds may result in some pages with no printing on them because the page falls entirely within a “whitespace” area of the map. By default, all pages are printed, even blank pages. To save paper, we can uncheck Print blank pages in multipage printouts in Tools - Options - User Interface.
File - Print
The contents of any component window may be printed in a default way with File - Print. The component will be printed on a single sheet (or multiple sheets in the case of text components using a simple dialog to set up the page. For much greater control over the appearance of a print job, use a Layout for printing.

The File - Print command calls the standard Windows Print dialog:

- **General tab** Choose the printer to use, the page range to print, collation and the number of copies.

The Printing Preferences dialog invoked by pressing the Preferences button in the Print dialog includes the following additional tabs (in older Windows versions these tabs may appear in the Print dialog itself):

- **Layout tab** Choose paper orientation from Portrait or Landscape. Includes an Advanced button to access any advanced configuration options available for the selected printer.

- **Paper/Quality tab** Choose paper tray selection. Also includes an Advanced button to access any advanced configuration options available for the selected printer.

See Also

How to Print

Layouts
File - Exit
Close all windows, close the project and exit Manifold System. Manifold will ask if we want to save the project if it has changed since the last save.

Edit Menu
Edit - Assign Projection
The Edit - Assign Projection dialog shows the coordinate system (projection) settings Manifold uses to interpret the coordinate numbers currently within a drawing or used with the pixels in an image. Changes in this dialog change merely the interpretation of the numbers and not the actual coordinate numbers or arrangement of pixels.

This dialog is also used to assign a projection to be used by a map. In this case, it tells the map what projection to use to display its contents. If the component layers in the map use different native projections, the map will re-project the components on the fly to show them as they would appear in the projection assigned to the map.

Note: This dialog is normally called just once when importing drawings or images from insensate formats. It is not the dialog to use to “re-project” a drawing or to specify a projection desired for a map. To re-project a drawing, image, or other component, use the Edit - Change Projection dialog.

The only time this dialog is used with components other than maps is to either verify the projection is correct or to specify the correct parameters after importing a projected component from a legacy format that does not contain projection parameter information. The component is imported as though it was unprojected (no parameters involved) and then this dialog is used to manually set the correct coordinate system parameters. Do not make changes in this dialog unless you are sure that the component was imported from a legacy format and now needs to have the correct coordinate system / projection parameters entered manually.

Although this dialog is visually similar to the Edit - Change Projection dialog (because both deal with similar data), the two are critically different in function. The Edit - Assign Projection dialog tells the system how to interpret existing data without modifying the data. The Edit - Change Projection dialog changes the data to fit a required interpretation.

The Edit - Assign Projection dialog is used to make changes on a “one time” basis only when an import from a legacy format requires manual entry of the correct projection parameters. Changing parameters in the Edit - Assign Projection dialog changes only the interpretation of the existing data and not the coordinate data.

The Edit - Change Projection dialog, in contrast, changes both the actual coordinate numbers as well as their interpretation. It is used every time one desires to recast a drawing or image into a new projection.
The dialog includes these controls:

- **Load from File** - Load projection information from a file using ERDAS AUX, Golden Software GSR, ESRI PRJ, or XML files.

- **Save to File** - Save projection information to a file using XML, Golden Software GSR or ESRI PRJ files.

- **Recent settings** - Choose a projection setting to reuse from a list of recently-used projections employed in this Manifold session.

- **Load from Component** - Load projection settings from another component in this project.

- **(Projection)** - A tree diagram providing numerous projections within various types of projections.

- **(Datum)** - A list of standard datums that incorporate various Earth ellipsoids and standard offsets.

- **(Parameters)** - One or more optional parameter settings, if used by this particular projection. Double-click into the values boxes to change them. Press Enter after changing a value.

- **(units box)** - Choose the unit of measure. All "unprojected" maps are in Degrees, Arc Minutes or Arc Seconds. Most projected...
maps are in meters.

**Local offset**
Shifted version of the external coordinate system peculiar to this component. Specifies shift in X (longitude) and Y (latitude) direction in the given units. With images or surfaces, the Local offset stores the offset to the left-bottom corner of the left-bottom pixel of the image or surface.

**Local scale**
Scaled version of the external coordinate system peculiar to this component. Specifies scale factor to be applied to external coordinate system.

**Scale correction**
Controls scale of the external coordinate system. For example, 0.9996 for UTM.

**False easting / northing**
Shift of the external coordinate system. For example, 500,000 meters for UTM. Some projections (such as UTM) add values to the "true" X and Y coordinates so that all coordinates stored are positive numbers. This avoids the use of the minus sign in primitive formats. The value added to X is the Easting and that to the Y is the Northing.

**Adjust for units**
Automatically adjust easting/northing values when units of measure are changed. Not checked by default.

**Clip coordinates**
Not used in the Edit - Assign Projection dialog.

**Autosuggest local values**
Not used in the Edit - Assign Projection dialog.

**Preserve local values**
Normally, choosing a projection from the projections pane will reset the values of Local offset, Local scale and units to the preset values for that projection. Checking this box will preserve any existing values in these local fields even when a new projection is selected. It is checked by default in the Edit - Assign Projection dialog because this dialog is often used to manually specify projection information when importing from geographically unaware legacy formats. In that case, any custom specifications for local values should probably be preserved when changing the projection.

**Suggest**
Suggest values for projection parameters based on the view in the component window. Used with projections with user-specifiable parameters.

The most frequent use of this dialog is to manually provide projection information when importing from formats too geographically unaware to provide such information (such as .shp or .dxf). After importing projected drawings from such files, the user must provide the missing projection information using the Edit - Assign Projection dialog.

To import a projected drawing or image from a legacy format:

1. Import the drawing or image using File - Import. Use default settings.
2. Open the drawing or image.
3. Use the Edit - Assign Projection dialog to specify the projection information that should be used.

The Edit - Assign Projection dialog allows us to manually provide the correct projection information into the coordinates properties. Manifold will then be able to make sense of the data just imported.

Do not confuse the assignment of a chosen projection view in a map window (via Edit - Assign Projection) with the use of Edit - Assign Projection in a drawing window to specify projection information missing from legacy formats. If a projected drawing is imported from, say, .shp format and you fail to tell Manifold the correct projection to use, that drawing has not yet been correctly imported. Subsequent use of the drawing in maps may
cause bizarre effects or lengthy delays as the system attempts to compute a new projection for the map based on fundamentally inaccurate coordinates.

**Note:** Maps can work faster if the drawings and images they contain as layers use the same projection that is requested of the map. If we always use a particular projection within a map, it will be worth it to re-project the constituent drawings and images of that map into that particular projection as well. To permanently re-project a drawing or image, use the Edit - Change Projection dialog.

**Adjust for units Option**

When manually changing units of measure the false easting and northing values will not be changed correspondingly. Checking the **Adjust for units** box will force a recomputation of the easting and northing values into the corresponding values for the new units of measure. For example, to change from Feet to Meters as units of measure and to change easting / northing accordingly:

- Verify the **Adjust for units** option is off,
- Set the units box to **Foot**,
- Turn on the **Adjust for units** option,
- Set the units box to **Meter**.

**Projection Toolbar Buttons**

We may want to re-use the same projection information with different components. For example, suppose we have many shapefiles covering the same region of interest that contain projected information requiring the use of the **Edit - Assign Projection** dialog after import. Since all of the files will require the same projection information, after importing one file and manually specifying the projection settings in the **Edit - Assign Projection** dialog we can save the projection information using the **Save to File** toolbar button. With subsequent files we can then use the **Load from File** button to load the settings and thus avoid manual re-entry of repetitive information.

Another time saver is to load projection information from an existing component using the **Load from Component** button. For example, we may want to re-project all files into a certain standard projection used by a given component and would like to be sure that all settings are identical. In that case, we open each component to be re-projected, launch the **Edit - Change Projection** dialog and use **Load from Component** to load the desired settings from the component we wish to use as the standard.

**Inadvertent Usage**

With components other than maps, the **Assign Projection** dialog should be used only once: when the component is first used to verify the projection is correct or to assign the correct projection if it was not correct. Using this projection in other circumstances is almost certainly a mistake.

In particular, it does not make sense to first verify the projection using the **Assign Projection** dialog, then change the projection using the **Change Projection** dialog and then finally to use the **Assign Projection** dialog. Once the projection is verified to be correct and then the projection is changed, it simply does not make sense to, in effect, state that the result is now wrong by attempting to use **Assign Projection** once more.

If we do attempt to use **Assign Projection** after doing a **Change Projection**, then Manifold will raise a warning message. If we elect to proceed, Manifold will allow us to continue as there are rare circumstances when expert usage may indicate a very minor change to a projection using **Assign Projection** after a **Change Projection** has been done. If we are in the habit of doing such exotic and expert things we can turn off the warning message by un-checking the **Confirm re-assigning projection after projecting data** option in the Tools - Options dialog.

**Additional Reading**

Experts may jump directly to the Projections Quick Reference topic. See the **Edit - Change Projection** topic for a discussion of the **Local scale**, **Local offset**, **Scale correction** and **False easting/northing** options. These are expert level parameters intended for professional use.
If you are new to coordinate systems in GIS or would like a refresher on coordinate systems and projections, consult the Coordinates Tutorial, the Projections Tutorial, and the Coordinates in Projected Maps topics followed by a quick review of the Projections Quick Reference topic for a summary.

See an example of usage in the Manually Georegister an Image example topic, in the Import a Projected Shapefile topic and in the Download and Mosaic Terraserver Images topic.

**Differences between Edit - Assign Projection and Edit - Change Projection**

- **Edit - Assign Projection** changes only the view of the data in maps and the interpretation of the data in other windows. It does not change the actual coordinates in the data. With map windows, it changes the projection used in the view without actually changing the data in the components. Map windows have dynamic projections that can show their contents in whatever projection is desired regardless of the native projections of what is inside them. When used with a drawing, image, labels or surface window it changes the interpretation of the data. This is a rarely used function that is utilized only to manually adjust projection properties when importing from formats that do not correctly store projection information.

- **Edit - Change Projection** appears with individual component windows such as drawing, image, labels or surface windows. This command changes the actual data in the component into a new coordinate system. There is no ability in a drawing window to see the drawing in a projection other than its native projection. We use **Edit - Change Projection** to re-project a component.

**Notes**

When opened with a read-only component (such as a drawing or other component stored on an Enterprise server when using Enterprise Edition), projection dialogs will report projection information but will not allow changes.

The **Assign Projection** and **Change Projection** dialogs also appear in the project pane context menu (right click on a component and the pop-up context menu appears). These dialogs will not be enabled in project pane context menus for components that take their projections automatically from parent components, for example, labels components bound to drawings or profiles.

**See Also**

**Edit - Change Projection**
Edit - Change Projection

The Edit - Change Projection dialog permanently changes the projection used for a drawing or image. With drawings, it changes the coordinate numbers within the drawing to their equivalents in the new projection, and then updates Manifold's coordinate properties for that drawing. With images or surfaces, it will re-compute the image by interpolating to add or delete pixels as necessary to match the image shape to the required projection.

To use the Change Projection dialog we must have first verified the projection currently assigned to the component to be re-projected. We do that by launching the Edit - Assign Projection dialog, making sure the existing projection is correct and then pressing OK. If we attempt to Change Projection without first verifying the projection, Manifold will not allow us to change the projection.

The dialog opens with a display of the current projection in use. Changing values in this dialog and then pressing apply will cause Manifold to re-compute the coordinate numbers within the drawing into their equivalents in the new projection. It is a permanent way of casting the data that define the drawing or image into a new projection.

The dialog includes these controls:

- **Load from File** - Load projection information from a file using ERDAS AUX, Golden Software GSR, ESRI PRJ, or XML files.

- **Save to File** - Save projection information to a file using XML, Golden Software GSR or ESRI PRJ files.
Recent settings - Choose a projection setting to reuse from a list of recently-used projections employed in this Manifold session.

Load from Component - Load projection settings from another component in this project.

(Projection) A tree diagram providing numerous projections within various types of projections.

(Datum) A list of standard datums that incorporate various Earth ellipsoids and standard offsets.

(Parameters) One or more optional parameter settings, if used by this particular projection. Double-click into the values boxes to change them. Press Enter after changing a value.

(units box) Choose the unit of measure. All "unprojected" maps are in Degrees, Arc Minutes or Arc Seconds. Most projected maps are in meters.

Local offset Shifted version of the external coordinate system peculiar to this component. Specifies shift in X (longitude) and Y (latitude) direction in the given units. With images or surfaces, the Local offset stores the offset to the left-bottom corner of the left-bottom pixel of the image or surface.

Local scale Scaled version of the external coordinate system peculiar to this component. Specifies scale factor to be applied to external coordinate system.

Scale correction Controls scale of the external coordinate system. For example, 0.9996 for UTM.

False easting / northing Shift of the external coordinate system. For example, 500,000 meters for UTM. Some projections (such as UTM) add values to the "true" X and Y coordinates so that all coordinates stored are positive numbers. This avoids the use of the minus sign in primitive formats. The value added to X is the Easting and that to the Y is the Northing.

Adjust for units Automatically adjust easting/northing values when units of measure are changed. Not checked by default.

Clip coordinates Enabled for projections like Orthographic: clip those parts of objects that extend past the projection extent. Manifold will clip coordinates to an accuracy of 0.01 degrees.

Autosuggest local values Enabled for images and surfaces: Automatically adjusts the values of local scale and local offset parameters after we change the coordinate system or alter one of the parameters in the parameter list. Checking Preserve local values overrides this box and keeps local values as they are.

Preserve local values Normally, choosing a projection from the projections pane will reset the values of Local offset, Local scale and units to the preset values for that projection. Checking this box will preserve any existing values in these local fields even when a new projection is selected. It is not checked by default in the Edit - Change Projection dialog because usually when re-projecting a component we want to use the preset local values associated with a new projection.

Suggest Suggest values for projection parameters based on the view in the component window. Used with projections with user-specifiable parameters.
Maps can work faster if the drawings and images they contain as layers use the same projection that is requested of the map. If we always use a particular projection within a map, it will be worth it to re-project the constituent drawings and images of that map into that particular projection as well. To permanently re-project a drawing or image, use the Edit - Change Projection dialog. In order for maps to run faster when the constituent drawings or images are in the same projection, they must use that same projection in all parameters. It does not help if both the map and its constituent drawings use Lambert Conformal Conic projection but with different center latitudes and longitudes.

Projection Toolbar Buttons

We may want to re-use the same projection information with different components. For example, suppose we have many shapefiles covering the same region of interest that contain projected information requiring the use of the Edit - Assign Projection dialog after import. Since all of the files will require the same projection information, after importing one file and manually specifying the projection settings in the Edit - Assign Projection dialog we can save the projection information using the Save to File toolbar button. With subsequent files we can then use the Load from File button to load the settings and thus avoid manual re-entry of repetitive information.

The Load from File button also provides a convenience when importing components created by other software packages. Using Load from File - Load projection information from a file using XML, Golden Software GSR or ESRI PRJ files.

Another time saver is to load projection information from an existing component using the Load from Component button. For example, we may want to re-project all files into a certain standard projection used by a given component and would like to be sure that all settings are identical. In that case, we open each component to be re-projected, launch the Edit - Change Projection dialog and use Load from Component to load the desired settings from the component we wish to use as the standard.

Sizing Images and Surfaces

When choosing Edit - Change Projection to re-project an image or surface the image size will often change slightly in size. The actual size of the resulting image or surface in pixels will be determined by the Local Scale parameters, which set by implication the size of each pixel. Choosing inappropriate scale parameters can result in absurdly large images or surfaces. Attempting to create a very large image or surface in this way will cause Manifold to raise a confirmation dialog.

Local Scale controls the image or surface resolution. A value of 5 meters means that each pixel of the resulting image will occupy an area of 5 meters by 5 meters. The greater the Local Scale, the fewer pixels the image will have. Local Offset specifies the offset to the left-bottom corner of the left-bottom pixel of the image or surface. When re-projecting images we can set the Local Scale. The system will compute Local Offset automatically.

Adjust for units Option

When manually changing units of measure the false easting and northing values will not be changed correspondingly. Checking the Adjust for units box will force a recomputation of the easting and northing values into the corresponding values for the new units of measure. For example, to change from Feet to Meters as units of measure and to change easting / northing accordingly:

- Verify the Adjust for units option is off,
- Set the units box to Foot,
- Turn on the Adjust for units option,
- Set the units box to Meter.

Discussion

Although this dialog is virtually identical to the Edit - Assign Projection dialog, the two are different in function. The Edit - Assign Projection dialog tells the system how to interpret existing data without modifying the data. The Edit - Change Projection dialog changes the data to fit a required interpretation.

The Edit - Assign Projection dialog is used to make changes on a "one time" basis only when an import from a legacy format requires manual entry of the correct projection parameters. Changing parameters in the Edit - Assign Projection dialog changes only the interpretation of the existing data and not the coordinate data.
The **Edit - Change Projection** dialog, in contrast, changes both the actual coordinate numbers as well as their interpretation. It is used every time one desires to recast a drawing or image into a new projection.

### Local and External Coordinate Systems

The offset, scale, scale correction and easting/northing factors are intended for expert use. These parameters are scale and offset values for use with either the local coordinate system or the external, projection coordinate system.

In a nutshell, the external coordinate system is a coordinate system specified by the projection and projection parameters as described in the usual treatises on projection systems. The internal or local coordinate system is the coordinate system peculiar to the component, which may be a scaled and shifted version of the external coordinate system. Local systems occur when working with images imported from certain formats and possibly with drawings as well.

The **Local scale** and **Local offset** values expose the internal values used by a component to control the scale and shift of the internal coordinate system. These are given in the projection's units of measure displayed in the units box.

The **Scale correction** and **False easting/northing** parameters control the scale and shift of the external coordinate system. "False easting/northing" is used instead of "offset" for historical reasons. These factors are commonly met in projection systems like UTM. **Scale correction** values are given in dimensionless units (such as 0.9996 for UTM) and **False easting/northing** values are given in degrees or meters depending on type of the coordinate system (such as 500,000 meters for UTM).

**Example:** When importing a Manifold System Release 4.50 map that uses a **Unit2Degree** factor of 10,000 (the default), the **Local scale** will be 0.0001 for both **X** and **Y**. This means that one native unit within the imported drawing equals 0.0001 degree. Modern Manifold editions still use full 1:250-millionth of a meter precision, but they will import the unit numbers from a 4.50 map knowing that the units used in that drawing had a scale factor of 0.0001.

**Example:** Increasing the **Local scale** in **X** of an image or drawing used as a layer in a map will make it look wider. Increasing the **Scale correction** in **X** will make it look narrower.

**Example:** Increasing the **Local offset** in both **X** and **Y** of an image or drawing used as a layer in a map will move the component towards the upper-right. Increasing the **False easting/northing** in both **X** and **Y** of an image or drawing used as a layer in a map will move the component towards the bottom-left (hence, the word "false").

**Local scale** and **Local offset** are applied to internal coordinates in a mirror image, inverse way as compared to how **Scale correction** and **False easting/northing** are applied to external coordinates within a projection system. In general, one can achieve exactly the same scale or offset effect by applying inverse values to either the local or the external scale/offset pairs.

The choice of which to use depends on the context of the activity. **Local scale** and **Local offset** values are better suited for manual georegistration while **Scale correction** and **False easting / northing** values are better for the professional use of projections.

### Differences between Major Projection Dialogs

There are two main projection dialogs that we use most frequently in Manifold.

The **Change Projection** dialog is used to change the projection of a component from one projection into another. It can only work correctly if the component already has been accurately assigned a projection. The **Assign Projection** dialog is used to verify and assign projections. It is used mainly to assign projections to components imported from formats that do not correctly store projection information. It is also used to assign the projection to be used by a map component.

- **Edit - Change Projection** is used to re-project a component into a different projection. This command changes the actual coordinate data in the component into a new coordinate system and also updates the projection assigned to that new projection. It appears with individual component windows such as drawing, image, labels or surface windows. It does not appear with map windows. We can only use **Change Projection** to change the projection of a component after we have verified that the projection assigned to that component is indeed correct using the **Edit - Assign Projection** dialog.
The assigned.

Why are there Two Projection Dialogs?

The need to have two projection dialogs that are confusingly similar to beginners, the Assign Projection dialog and the Change Projection dialog, arises from the GIS industry's inability to invent a clean and simple solution to the general use of projections. That the industry has been unable to invent a clean and simple solution is at the same time a negative sign and it is also a very positive sign about the health of the industry.

It is a positive sign because the GIS industry has become very large and a modern GIS like Manifold can read data from a vast number of formats used to store images, drawings and other data. Many sources for such data don't have anything to do with the GIS industry and so don't have the technical awareness that, say, a particular image might be used some day in a GIS package. Such sources don't bother implanting projection information into that image and might not even know what a projection is or how to correctly store such information.

For example, one day we may be flying in an airplane over an interesting view, and we might take a snapshot of that view with an ordinary consumer digital camera. The makers of the camera don't know anything about GIS and they have no way of knowing that a camera normally used to take snapshots of pets and birthday parties might also be used one day to take an aerial photograph that someone would like to use in a GIS package like Manifold. It is a testament to the power of GIS and to the power of Manifold that even though this image was acquired using non-GIS technology Manifold nonetheless has facilities (such as georegistration or the use of the Assign Projection dialog) that let us use that image in Manifold. Very cool.

The negative scenario driving the need for two dialogs is the very wide distribution within the GIS community of lots of data stored in absolutely terrible, poorly-designed formats that do not store projection information accurately. In years past many GIS packages were simply awful from a software design perspective and lacked the foresight to design a well-functioning format. In some cases, the old formats are limited because the equipment used to acquire the data was designed decades ago: for example, space technology changes so slowly that many of the sensors operating on today's satellites for space imagery were designed many years ago. They might have even been designed by people who never heard of DVDs, cell phones, Windows or the Internet.

There is an incredible amount of legacy junk out there in GIS, and some of it is still earnestly advocated by people who don't realize it is dreadfully obsolete. If you use a modern GIS like Manifold, to make an analogy with the music world, it is as if you are used to the modern world of MP3's and CDs but when you go down to your local
government GIS department you encounter a data librarian who thinks the world still uses eight-track audio tape cartridges and offers you one of those. While you stare at the antique thing thinking "you've got to be kidding..." the librarian is beaming with pride over what he or she thinks is modern technology.

To be absolutely fair, the widespread distribution of GIS data in junk, archaic formats is in some ways a positive sign: it shows that GIS has been around long enough for governments and organizations to acquire a really titanic amount of data about our world already stored in some sort of GIS format. That's very useful because getting that into Manifold is usually easy if you pay attention to some small details. It's also positive in that we wouldn't think of such archaic formats as being utter junk if the GIS industry were not evolving so rapidly that new things like Manifold appear to make the old stuff seem so old-fashioned. But it still complicates matters for beginners.

How Projections Work

To understand why we need two confusingly similar projection dialogs we need to understand how projections are made to work. Bear with us and do not skip over this section as it is critically important to understanding how to operate Manifold or any other GIS.

When we pop open a drawing the system draws pretty lines and areas and points based upon data sitting in a file somewhere. That data consists of a series of paired numbers that give the locations of coordinates needed to draw what we see in a "connect the dots" fashion. If you zoom far enough into any drawing you'll see that even though the drawing may appear to have smooth curves, when viewed close up it always consists of straight line segments that are drawn between coordinate locations.

To make this system work, where a file with a bunch of numbers in it is displayed as a pretty drawing, there are two things that the GIS system has to know how to do:

- It has to have some coherent scheme for representing coordinate locations as numbers, and
- It has to know how to display those numbers as objects in the drawing.

Obviously, whoever designs the GIS package better make sure that whatever scheme is chosen for coordinate numbers is something that is understood by the module used for display.

Software designers can come up with all sorts of ways of storing coordinates as numbers in files and indeed there are many ways of doing so. A simple package could store numbers that specify latitudes and longitudes of coordinates. A CAD package like AutoCAD could store numbers that represent some arbitrary X,Y grid position in a blueprint. A technical cartographic package might store numbers like 43080934098.3048939 that represented some grid location within a projected coordinate system.

It almost doesn't matter what numbering system is used to encode the coordinates so long as the rest of the GIS package knows how to use those numbers to display objects in the drawing and to work with them. The key thing is that the part of the software that works with those stored coordinate numbers must understand what they are supposed to mean. It is as if the drawing were stored in one language, English, and the display software had to understand that English was used as well.

To continue this analogy, it doesn't matter if the drawing were stored in Japanese if the display software also understood it was Japanese. That would OK as well. But, a situation where the drawing was stored in Japanese and the display software thought it were English would be a problem, or if it were stored in English and the display software thought it was in Japanese would also be a problem. The key thing is that both parts of the system must agree on what is supposed to be going on.

A more technical reader at this point may jump ahead to think, "Hey, all that is great, but if here I sit using Manifold, why do I care what other systems might do? Why is it simply not the case that Manifold picks some way of storing numbers, understands within itself what it is doing, and we can then forget about any of these complications?" Good question.

Manifold can and does indeed have an internal format for storing coordinates that all the other parts of Manifold understand perfectly. In a perfect world we'd never have to worry about any of this stuff. There are two catches to why we still need two projection dialogs and why Manifold's internal system does not simplify life as we would like:

- We have to get data from other systems into Manifold, and other systems have all sorts of different ways of storing numbers for drawings. Sometimes that's easy to deal with, but sometimes we have to figure out how the other system stores coordinate numbers and what it intended. For this reason...
(and for others) we need a dialog that we can use to tell Manifold what is intended. That's why we have an Assign Projection dialog.

- Because of the limits of hardware speed, even within Manifold itself just one way of storing coordinate numbers is not fast enough to allow us to use many different projections. Sometimes we have to use different coordinate number schemes to get the performance we want. For this reason, we need a dialog that we can use to tell Manifold to change the numbers into a different scheme. That's why we have a Change Projection dialog.

The first point above will get beaten to death in endless essays and topics throughout this documentation. Reading those topics will help the Manifold user get street-smart about exploiting data acquired in dumb formats. We won't go further into it here.

The second point is more interesting and often comes as a surprise to non-technical users. In modern times we all are used to computers storing things internally in various magic ways without needing to know how all that technical wizardry is accomplished, so it may come as a surprise to have to think about how that is done. But that happens in GIS because GIS data sets are often so huge that even with modern desktop computers we have to make some manual compromises to be able to work with such data.

It's a bit like having an automatic transmission in a car as opposed to a manual transmission: if you have a really weak engine in a car but still have to carry heavy loads you might need to learn how to use a clutch and how to manually shift gears to eke out every last bit of capability. In the GIS world, sometimes to deal with the limits of processing on desktop computers we have to know how to manually "shift gears" by re-projecting a drawing or other component.

If all computers were super-fast supercomputers, we could always store all coordinate numbers using the same scheme, for example, as simple latitude and longitude numbers, and then whenever we needed to show them in some projection we could tell Manifold what projection to use and Manifold would make the calculations on the fly to show the data in the desired projection. No matter how big the drawing might be, no matter how many numbers may be involved, if a fast enough supercomputer were doing the work Manifold could instantly show the drawing to us in whatever projection we wanted. That would be like an automatic transmission in a car.

In fact, computers have gotten fast enough that Manifold can often do just this when we show drawings or other components in a map window if the drawings or images are not too large. If the drawings use a different projection (that is, their data is stored using a different number scheme) than the projection used by the map, the map will actually re-compute the display on the fly so that the drawing layers appear in that map as if they used the map's projection as their own native projection.

The problem is that only goes so far. When the drawing gets big enough, the time required to re-compute the drawing's numbers into a different projection takes so long that users get annoyed while their maps take forever to display. Most GIS images are already so big that users don't want them re-projected on the fly. If we are cursed with the bad luck of not having a supercomputer on our desktop we will want the display process to go faster and to do that we will have to change projections manually sometimes, like using a stick shift in a car.

The way to make the display process go faster is to change the numbers inside the drawing into a numbering scheme that directly implements the desired projection. On a one-time basis we can re-compute the coordinate numbers inside the drawing so that it later can be displayed in the desired projection without any need to re-compute it on the fly. It's true we have to wait while the drawing is re-projected, but that only happens once and then every time we display it in that projection it displays fast. This is what the Change Projection dialog does. It changes the innards of a drawing or other component so that the native projection is the one we desire, the one in which we expect to work with the drawing.

All the above notions set up a two stage methodology involving projections. We have to know what the numbers are used inside any given drawing, image or other component. The numbering scheme used internally is the native projection used inside the drawing. And then also Manifold has to be told what projection is supposed to be used with those numbers, which is the assigned projection. Obviously, the assigned projection should match the native projection.

All Manifold features and functions will automatically make sure that happens, assuming that the data set was correctly set up (that is, the assigned projection matched the native projection) in the first place. If we use the Change Projection dialog to change the native projection of a drawing, Manifold will not only change the native projection but it will also change the assigned projection to match as well. We can therefore use Change Projection to re-compute the numbers (that is, to re-project the numbers) inside a drawing into whatever projection we want for fastest possible work. Inside Manifold, it all works automatically and perfectly with nothing to worry about.
But when we bring something into Manifold from outside, we may need to tell Manifold what those numbers are supposed to mean. In that case, we will use the **Assign Projection** dialog to tell Manifold what projection to assign to that drawing so that the assigned projection matches the numbering scheme used inside the drawing.

You might think that when Manifold imports a drawing or image it should know what the numbers are supposed to mean. If we are importing from a modern GIS format that correctly stores projection information that will happen and we won’t have to worry about using **Assign Projection**. But if we are stuck importing from some stupid format like DXF or ESRI shapefiles, then we will have to use **Assign Projection**. That’s life in the big city.

You might think that anyone with common sense, no matter how much a beginner at programming, would make sure that any format they designed for storing GIS data would also embed some information as to what the numbers are supposed to mean, but unfortunately the programmers who designed many of the world’s GIS legacy formats lacked the foresight to do so. So, we are stuck with hacks like shapefiles filling up the world’s hard disks and stuck with the need to manually use **Assign Projection** to import such things. At least we can be grateful that Manifold has tools to let us import such data almost no matter what format it is in.

**Two More Uses for Assign Projection**

OK, so we use **Change Projection** to re-compute the numbers inside a component and we can trust Manifold to also assign the right projection to match that re-computation when it is done. If we bring something into Manifold from some wacko format and need to tell Manifold what those numbers mean, we use the **Assign Projection** dialog. So far, so good.

In addition to the above, we also use the **Assign Projection** dialog in two related uses that are so similar in purpose that we can use this same dialog for them:

- **To verify** the projection that has been assigned to a component. Manifold won’t trust that a drawing or other component has been imported correctly with the right projection assigned, even if we import the item from some smart format. Instead, Manifold insists that we take a quick look at the projection that was assigned by popping open the **Assign Projection** dialog and pressing **OK**. This verification step is required to protect beginners from themselves, because if the correct projection is not assigned on import then chaos will ensue. Experts can turn this verification requirement off in the **Tools - Options** dialog.

- **To assign** the projection that a map is supposed to use. Although a GIS expert may quibble that this is a slightly different intent than what is discussed above, the concept is so similar (changing the way the data is interpreted rather than re-computing the data itself) that the **Assign Projection** dialog is used in this case as well. It’s the same information and dialog boxes required so we may as well use the same dialog.

**Notes**

When opened with a read-only component (such as a drawing or other component stored on an Enterprise server when using Enterprise Edition), projection dialogs will report projection information but will not allow changes.

The **Assign Projection** and **Change Projection** dialogs also appear in the project pane context menu (right click on a component and the pop-up context menu appears). These dialogs will not be enabled in project pane context menus for components that take their projections automatically from parent components, for example, labels components bound to drawings or profiles.

**Additional Reading**

Experts may jump directly to the Projections Quick Reference topic.

If you are new to coordinate systems in GIS or would like a refresher on coordinate systems and projections, consult the Coordinates Tutorial, the Projections Tutorial, and the Coordinates in Projected Maps topics followed by a quick review of the Projections Quick Reference topic for a summary.

See also the Projections and Images topic.

**See Also**

**Edit - Assign Projection**
Edit - Change Projection
Edit - Undo / Redo
A limited form of Undo and Redo that provides single-step undo and redo capabilities in certain contexts when ambiguities can be resolved. Comments, queries and scripts allow multi-level Undo and Redo. Certain operations (for example, certain remote database operations) cannot be undone.

It is always wise, therefore, to make frequent saves and to always work with copies of important data in case changes get out of hand.

When working with large images and surfaces the Undo command requires the creation of large temporary files. The process of creating such temporary files can be quite slow and result in annoyingly slow performance when working with very large images and surfaces. Many users prefer to turn off Undo in such cases.

The parameters specified in Tools - Options - Confirmation allow users to the size of images and surfaces that preserve the ability to Undo. The Confirm creating undo actions larger than… option provides two controls:

- The ability to ask for confirmation before an Undo action is undertaken when image or surface size exceeds a given number of megabytes, by default, 50 megabytes.
- A standing specification to never undertake an Undo action larger than a given number of megabytes, by default, 250 megabytes. If this second parameter is reduced to a very small number of megabytes the effect will be to turn off Undo except for very small images and surfaces.

Note: Script windows, query windows and comments are unusual in that they support multi-level Undo / Redo. One can CTRL-Z (Undo) backwards through many changes and CTRL-Y (Redo) forwards to redo many Undo operations.

Tech Tip
Many transform operators for drawings automatically run a Normalize Topology transform before running to eliminate common errors that may affect correct operation. When such an operator is undone, the normalization that automatically preceded it will also be undone.
Edit - Cut / Copy

Either **Cuts** and copies or simply **Copies** selected items onto the Windows clipboard. Shortcuts are CTRL-X for Cut and CTRL-C for Copy.
Edit - Paste / Paste Append
Pastes contents of the Windows Clipboard into the active component. Edit - Paste Append appears in components such as drawings, tables or comments, where data may be appended onto existing text or records.

When copying objects from one drawing and pasting into another the Paste Objects dialog appears to allow control of how fields are pasted.

Paste Objects Dialog Controls

Select All - Paste all columns.
Select None - Do not paste any columns except ID.
Select Inverse - Do not paste all current Paste As columns and paste all other columns. A fast way to use all but one column: click Select None, specify a Paste As choice for the one column not desired and then click Select Inverse.
Autoselect - Paste existing fields to compatible columns where the names are the same and otherwise create a new field as necessary.

Column This column lists fields in the source drawing.
Paste As How the field should be pasted into the destination drawing. Double-click into a Paste As cell to

Pasting Drawings, Images or Surfaces as Tables
The Paste As dialog that appears when pasting drawings, images or surfaces as tables the columns offered will be intrinsic fields relevant to that component.

See Also

Intrinsic Fields in Tables
Project Pane - Paste as Surface
Project Pane - Paste as Drawing
Edit - Delete / Delete All
Appears with all components where items may be selected or highlighted.

Delete   Delete all items in the selection.
Delete All Delete all objects or pixels in the component. Will always ask for confirmation before deleting everything.
Menus, Dialogs and Controls Reference

**Edit - Select All / None / Inverse**
Changes selection throughout entire component at once.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select All</td>
<td>Select all objects or pixels in the component.</td>
</tr>
<tr>
<td>Select None</td>
<td>Deselect all objects or pixels in the component.</td>
</tr>
<tr>
<td>Select Inverse</td>
<td>Invert the selection. Those items that were not selected will become selected, while those that were selected will be deselected.</td>
</tr>
</tbody>
</table>

Select All and Select None are obvious in action, as seen above using an image as an example.

Select Inverse is a very useful command, because sometimes when making a selection it is easier to pick out those things we do not want to select. For example, if we ultimately would like to select the land area of South America in the images above, it is easier to first use Touch Select to select ocean areas and to then use Edit - Select Inverse to get the land areas. The ocean regions are very even in color and can be selected with a single click. The land areas contain many different colors and would take many clicks when using a reasonable tolerance setting with Touch Select.

These commands also work with drawing layers. In the sequence above, we would like to retain a circular region of lines in a drawing layer while deleting all other lines. We do this by selecting the circular region, choosing Select Inverse and then deleting the new selection (all the lines except the ones we want).
These commands also work with charts, tables and even forms.

For example, if one control in a form is selected, using **Select Inverse** will select the other controls.
Edit - Select Mode

The Edit - Select Mode menu provides menu access to selection mode buttons on the Selection toolbar. Selection mode buttons control how a new selection made with a selection tool is combined with any existing selection. One selection mode button is always pushed in.

- **Select Replace** - Replace the existing selection with whatever is now selected.
- **Select Add** - Add whatever is now selected to the existing selection.
- **Select Subtract** - Subtract whatever is now selected from the existing selection.
- **Select Invert** - Deselect whatever is now selected that was already in the existing selection and otherwise add what has just been selected.
- **Select Intersect** - Select only the region of overlap between what is specified by the tool and what is in the existing selection. If there is no overlap, nothing will be selected.

See the main Selection topic for details and examples.
Edit - Select Objects

The **Edit - Select Objects** menu appears with drawings to provide menu access to selection mode buttons on the
Selection toolbar.

Selection filter buttons allow us to specify what types of objects are selectable. This is very useful when working
with complex maps with lots of objects in sight, because it allows us to enable selections only for certain items,
such as points.

Choose the type of items that are to be selected by pushing the selection filter buttons for the desired types. The
buttons above show selection enabled for points and disabled for areas, lines and pixels. If we drew a selection
rectangle over a region in the map that contained all four types of items, only the points within the rectangle will
be selected. Areas, lines and pixels within the rectangle will be ignored.

![Select Areas](image1)

Select **Areas** - Enable selection of areas by mouse
selection tools.

![Select Lines](image2)

Select **Lines** - Enable selection of lines by mouse
selection tools.

![Select Points](image3)

Select **Points** - Enable selection of points by mouse
selection tools.

![Select Pixels](image4)

Select **Pixels** - Enable selection of pixels by mouse
selection tools.

![Select Text](image5)

Select **Text** - Enable selection of text labels by mouse
selection tools.

All four filter buttons may be used in maps. When working with drawings, the **Select Pixels** button will be
disabled. When working with images, **Select Area**, **Select Lines** and **Select Points** buttons will be
disabled. When working with images, **Select Pixels** is always enabled and will be shown in a "forced" style that
cannot be clicked out.

This menu appears only for drawings, because with images **Select Pixels** is always the only filter button enabled.

See the main Selection topic for more information on selections.
Edit - Select by Type

The Edit menu includes a Select by Type choice. Use this command to select all of one or more types of objects, that is, all areas, lines or points in the drawing. The Select by Type command applies whatever Selection Objects and Selection Modes settings are in force.

For example, to select all of the points in a drawing we would push in the Select Points button and the Replace Selection button and then choose Edit - Select by Type. All points in the drawing will be selected, replacing any previous selection.

If we simply wanted to add all the points to the current selection we would have pushed in the Add to Selection mode button and then chosen Edit - Select by Type.
Edit - Modify Selection

This menu item appears with images only. The Edit - Modify Selection menu provides four choices for modifying the shapes of regions of pixels that have been selected:

- **Border**: Take the current region of selection and convert it to region of selected pixels that are +/- the given width of pixels from the boundary of the current region.
- **Contract**: Shrink the region of selected pixels by the given number of pixels.
- **Expand**: Expand the region of selected pixels by the given number of pixels.
- **Smooth**: Create a smoother periphery of the selected region using the given parameters.

See the Modifying Selections topic for details and examples.
Masks are grayscale images that are used to control other images. Masks are used for two purposes:

- To make a selection in the shape shown by the mask.
- To alter the values in the R, G, B or alpha channels of the target image using the values in the mask.

A mask can be any grayscale image. There is nothing special about grayscale images used as masks. What makes a grayscale image a "mask" is simply how we use it. See the Selection using Masks and Masks topic for details and examples.

We can create masks using any method used to create or edit grayscale images. Since masks are often used as a means of saving selections or other patterns from existing images Manifold provides an Edit - Save Mask/Channel command that makes it easy to create a mask using an existing image.

These commands can also be used with surfaces.

**Choices for Save Mask/Channel**

Any of the following aspects of an image may be saved as a mask:

- The Selection.
- The region of invisible pixels.
- Any saved selection we have made for this image in the Selections pane.
- The R, G, or B channels in the image if it is an RGB image.
- The alpha channel in the image if it is an RGBA image.

**Note:** The Save Mask/Channel dialog will not list any saved selections that are empty. If no selection is present in the image the dialog will not include a Selection choice. If the dialog contains no invisible pixels there will be no choice for Invisible pixels available.

We can load an image using a mask as a guide by using the Edit - Load Mask/Channel command.

**Choices for Load Mask/Channel**

Using a mask we can load any of the following aspects of an image:

- The Selection.
- Make the pixels in a given region invisible pixels.
- The R, G, or B channels in the image if it is an RGB image.
- The alpha channel in the image if it is an RGBA image.

When used to specify the Selection or invisible pixels every pixel in the mask that is black signifies no selection or no invisible pixel at that spot. Any pixel that is not black signifies making a selection or an invisible pixel at that spot. When using a mask to specify a region of invisible pixels any previously invisible pixels will remain invisible since making a pixel invisible is always a "one-way" operation that deletes the pixel.

When used to specify an R, G, B or alpha channel the value from 0 to 255 of the grayscale pixel is used for the value of the specified channel for that pixel location.

**Load** Which part of the image is to be loaded. The selection, a channel, or invisible pixels.

**From** The source image to use for the mask. All grayscale images available will be listed together with their sizes.

**Mode** Specifies how to use the mask within the target image:
Add - Add the intensity values of the mask pixels to the target.

Invert - If the intensity of a mask pixel is greater than 128, invert the value of the target pixel.

Maximum - Use the maximum of intensity of either the mask pixel or the target pixel.

Minimum - Use the minimum of intensity of either the mask pixel or the target pixel.

Replace - The default. Use the values of the mask pixels.

Subtract - Subtract the intensity of the mask pixel from the target pixel.

Restrict to Selection - If checked (the default), when a selection is present the mask will be applied only within the selected pixels.

See the Masks topic for examples.

Use with Surfaces

The Edit - Save Mask/Channel dialog for surfaces includes choices to save a grayscale image, invisible pixels or aspect or slope data in single-precision floating-point format into new surfaces. Aspects vary from -180 to 180 degrees. Slopes are given in degrees. Surfaces created using this dialog will retain the coordinate system (projection) and saved selections of the original surface.
Edit - Find / Find Next

Appears with all components that contain text or have searchable data associated with them. Searches with regular expressions to match a custom pattern of characters are allowed if the Use regular expressions box is checked.

**Very important:** Searches proceed from the current record to the last record in the table. To begin at the beginning, we must press the Restart button.

Any located object found with Find or Find Next becomes the primary selected object and clears the selection.

<table>
<thead>
<tr>
<th><strong>Find what</strong></th>
<th>The value or regular expression to be found.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Match whole word only</strong></td>
<td>Match the value only when it occurs as a whole word. With this checked, &quot;the&quot; will match the word the but not occurrences of these letters within words such as weather.</td>
</tr>
<tr>
<td><strong>Match case</strong></td>
<td>Find is case insensitive by default. This forces a match exactly on the case given in the Find What value. Example: unchecked, &quot;the&quot; will match The and The.</td>
</tr>
<tr>
<td><strong>Use regular expressions</strong></td>
<td>Regular expressions are a concise and flexible notation for finding and replacing patterns of text.</td>
</tr>
<tr>
<td><strong>Search only in</strong></td>
<td>By default, Find will search all fields. Check this to restrict the search to only the field given.</td>
</tr>
<tr>
<td><strong>Find Next</strong></td>
<td>Find the next instance of the value.</td>
</tr>
<tr>
<td><strong>Find All</strong></td>
<td>Find and select all instances of the value.</td>
</tr>
<tr>
<td><strong>Restart</strong></td>
<td>Move the current record pointer to the beginning of the table. Searches proceed from the current record pointer only.</td>
</tr>
<tr>
<td><strong>Close</strong></td>
<td>Close the dialog</td>
</tr>
</tbody>
</table>

See the Regular expressions topic for information on using regular expressions. Regular expressions are a sophisticated and flexible way of searching for patterns of text.
Edit - Replace

Appears with all components that contain text or have searchable data associated with them. Searches with regular expressions to match a custom pattern of characters are allowed if the Use regular expressions box is checked.

Very important: Searches proceed from the current record to the last record in the table. To begin at the beginning, we must press the Restart button.

Any located object replaced with Replace or Replace All becomes the primary selected object and clears the selection.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Find what</td>
<td>The value or regular expression to be found.</td>
</tr>
<tr>
<td>Replace with</td>
<td>Value used to replace found values.</td>
</tr>
<tr>
<td>Match whole word only</td>
<td>Match the value only when it occurs as a whole word. With this checked, &quot;the&quot; will match the word the but not occurrences of these letters within words such as weather.</td>
</tr>
<tr>
<td>Match case</td>
<td>Replace is case insensitive by default. This forces a match exactly on the case given in the Find What value. Example: unchecked, &quot;the&quot; will match The and tHe.</td>
</tr>
<tr>
<td>Use regular expressions</td>
<td>Regular expressions are a concise and flexible notation for finding and replacing patterns of text.</td>
</tr>
<tr>
<td>Search only in</td>
<td>By default, Replace will search all fields. Check this to restrict the search to only the field given.</td>
</tr>
<tr>
<td>Find Next</td>
<td>Find the next instance of the search value.</td>
</tr>
<tr>
<td>Replace</td>
<td>Replace this instance of the search value.</td>
</tr>
<tr>
<td>Replace All</td>
<td>Replace all instances of the search value.</td>
</tr>
<tr>
<td>Restart</td>
<td>Move the current record pointer to the beginning of the table. Searches proceed from the current record pointer only.</td>
</tr>
<tr>
<td>Close</td>
<td>Close the dialog</td>
</tr>
</tbody>
</table>

See the Regular expressions topic for information on using regular expressions. Regular expressions are a sophisticated and flexible way of searching for patterns of text. Regular expressions can also be used for specifying the Replace with text.
Edit - Advanced

The Edit - Advanced menu becomes available when editing text components such as comments, scripts and queries.

- **Make Lowercase**: Convert all characters in the selected text to lowercase.
- **Make Uppercase**: Convert all characters in the selected text to uppercase.
- **Comment Selection**: Comment out the selected lines of text.
- **Uncomment Selection**: Uncomment the selected lines of text.
- **Wrap Selection**: Appears for comments. Word wrap the selected text to the current window width.
- **Join and Wrap Selection**: Appears for comments. Word wrap the selected text to the current window width, deleting single line breaks to form single paragraphs.
- **Complete Word**: Appears for queries. Complete the word automatically if an unambiguous choice is possible. If no unambiguous choice is possible, pop up a list of choices (the same as Complete Word with List). Usually invoked with the standard Microsoft CTRL-space shortcut.
- **Complete Word with List**: Appears for queries. Pop up a list of possible completions. Not usually used, since checking (the default) the Automatically list completion suggestions when editing queries box in Tools - Options - Miscellaneous causes the list to pop up automatically.

Each command has a keyboard shortcut that is listed next to the command.

Comments

Manifold understands commenting conventions for a variety of languages, including SQL, VBScript, JScript, Python, PERL and .NET languages such as C#, VB .NET and JScript .NET. Comment style will be taken from the language used for a script or for SQL for a query.

Programmers will become familiar with the CTRL-K and CTRL-SHIFT-K sequences to comment and uncomment sections of code.

Join and Wrap Selection

This command joins paragraphs separated by a single line break but does not join paragraphs separated by two line breaks (that is, a blank line between paragraphs).

Consider a comments text window with three paragraphs, each separated from the others by two line breaks. We select the text and choose Edit - Advanced - Join and Wrap Selection.
The result is that each paragraph is wrapped to the current width of the window in use.

Consider the case where the same paragraphs are separated by only one line break at the end of each line. Once again we select the text and choose **Edit - Advanced - Join and Wrap Selection**.

Once again the text is wrapped, but this time the paragraphs are **joined** into a single paragraph.
Bookmarks are icons in the left margin of text components such as comments, queries and scripts. They allow fast navigation within lengthy text. Bookmarks are also available within tables and queries. Within tables they are very useful for marking particular records when jumping back and forth between records in a table. When tables or queries are sorted the bookmarks correctly travel with the records into the new sort order. If comments, queries or scripts are edited the affected bookmarks will be offset or removed as necessary.

The illustration above shows a bookmark icon in the left margin.

To set a bookmark:
1. Open the component.
2. Click on the line or record where the bookmark is to be placed. This positions the cursor on that line.
3. Press F8 or choose Edit - Bookmarks - Toggle from the main menu.

To clear a bookmark:
1. Go to the line or record marked with the bookmark to be cleared. (Use Edit - Go To or simply position the cursor on that line).
2. Press F8 or choose Edit - Bookmarks - Toggle from the main menu.

Note that the Toggle command sets a bookmark if one is not present and clears a bookmark if one is already present. The Edit - Bookmarks - Remove All command clears all bookmarks from the active component.

To go to a bookmark:
1. Choose the Edit - Go To command.
2. Choose Bookmark in the Go To box.
3. Choose the desired bookmark in the lower box and press OK.

The lower box provides a list of bookmarks shown using the line number and the first few characters of the line.
In the Go To dialog shown above we are about to jump to the bookmark at line 434 that is shown in the illustration at the beginning of this topic.

**Color Used**

Set the color used for bookmark icons in the Colors page in the Tools - Options dialog. Preferred bookmark color seems to be a very personal issue. The default is cyan to match Microsoft Visual Studio style. Some users prefer yellow or green.
**Edit - Breakpoints**

Breakpoints are icons in the left margin of script components that mark a program line at which execution automatically stops when the Manifold Debugger is in use.

Breakpoints are used together with the debugger to debug scripts. Breakpoints may be set at a given line to stop execution at that line. Once set, a breakpoint may be disabled to allow execution to flow past that line while still marking the line so it is easy to locate. Breakpoints appear in the left margin of scripts. If scripts are edited the affected breakpoints will be offset or removed as necessary.

The illustration above shows two breakpoint icons in the left margin. The lower round icon in the margin of the script shows a disabled breakpoint. Execution will flow past that line, although we can still use the breakpoint Go To Next / Previous commands or the Edit - Go To command to jump to that breakpoint and line quickly. The upper red icon in the margin is an enabled breakpoint. Execution will stop at that line.

- **Toggle** - Cycle the breakpoint at this line between "no breakpoint", "breakpoint set" and "disabled breakpoint."
- **Go to Previous** - Go to the previous breakpoint (includes both disabled and enabled breakpoints).
- **Go to Next** - Go to the next breakpoint (includes both disabled and enabled breakpoints).
- **Delete All** - Clear all breakpoints from this component.

<table>
<thead>
<tr>
<th>F9</th>
<th>Toggle</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHIFT-CTRL-F9</td>
<td>Go to Previous</td>
</tr>
<tr>
<td>CTRL-F9</td>
<td>Go to Next</td>
</tr>
</tbody>
</table>

**To set a breakpoint:**

1. Open the component.
2. Click on the line where the breakpoint is to be placed. This positions the cursor on that line.
3. Press F9 or choose Edit - Breakpoints - Toggle from the main menu.

**To cycle a breakpoint:**

1. Go to the line marked with the breakpoint to be disabled or cleared. (Use Edit - Go To or simply position the cursor on that line).
2. Press F9 or choose Edit - Breakpoints - Toggle from the main menu to disable the breakpoint.
3. Again press F9 or choose Edit - Breakpoints - Toggle from the main menu to clear the breakpoint.

Note that the **Toggle** command cycles breakpoint status for a given line between no breakpoint, breakpoint set and a disabled breakpoint. The **Edit - Breakpoints - Remove All** command clears all breakpoints from the active component.
To go to a breakpoint:

1. Choose the Edit - Go To command.
2. Choose Breakpoint in the Go To box.
3. Choose the desired breakpoint in the lower box and press OK.

The lower box provides a list of breakpoints shown using the line number and the first few characters of the line.

In the Go To dialog shown above we are about to jump to the disabled breakpoint at line 7 that is shown in the illustration at the beginning of this topic.

Colors Used

Set the colors used for breakpoint line background in the Colors page in the Tools - Options dialog. If the Highlight Entire Lines option is on, the breakpoint color will be used for the background color of the line containing the breakpoint. The default appearance is similar to that used in Microsoft's Visual Studio.

See Also

Debugger
**Edit - Go / Go To**

*Edit - Go To* provides a rapid way of panning and zooming to a desired view. Options include:

- **Address** Enabled if the Geocoding Tools extension has been installed. Go to a given street address when that address is provided either in full or partial form.

- **Layout** Enabled if a layout has been created. Pan and zoom to that layout.

- **Location (latitude / longitude)** Go to a given latitude, longitude coordinate.

- **Location (projected coordinates)** Go to a given coordinate using projected coordinates. Handy for use with UTM and similar systems.

- **Objects** Go to all objects (zoom to entire component), the selection or to a saved selection.

- **Zip code** Enabled if the Geocoding Tools extension has been installed. Go to a given ZIP code.

*Edit - Go* appears with tables only, and allows jumping the current record to the First, Next, Previous, Last or end position in the table to insert a new Blank record.

With script, comment and query windows, *Edit - Go To* can jump to any text line or bookmark if bookmarks have been inserted into the component. Go To in scripts can jump to any breakpoint if breakpoints have been inserted into the component.

To jump to a particular bookmark, choose it in the lower combo box and press OK. Bookmarks will be shown prefaced with their line numbers.

To jump to a breakpoint, choose Breakpoint in the Go To combo box and then choose the breakpoint desired. To jump to a line number, choose Line in the Go To box and then enter a line number in the lower box.
Geocoding Tools

If the Manifold Geocoding Tools extension has been installed the Edit - Go To command will allow a GoTo to an Address or to a Zip code. Area of coverage depends upon the geocoding data source that is being used. For example, using the Manifold geocoding database coverage will extend to the US only. If MapPoint is used coverage will be the North American or European coverage provided by the MapPoint edition in use.

The address may be a full street address, or it may be a partial address, such as "Atlanta" or "Atlanta, GA" or "GA." Manifold includes a gazetteer of large city names so that "Atlanta" will find the large city in Georgia and not one of the various small towns of that name throughout the US. City names will take priority over states when spelled out, so that "Washington" will find the capital city of the US and not the state. To find the state, use "WA."

See Also

Geocoding Tools
Geocoding Data Sources
**Edit - Snap To**

Checking a choice in the **Edit - Snap To** menu controls Snap modes, which work with mouse cursor motion. Snap modes force the mouse cursor to take effect only at those discrete, specified locations that are nearest to the mouse cursor's location. We can also set snap modes using Snap Toolbar buttons. When snap modes are engaged the status bar will show snap is turned on.

When any snap mode is engaged a secondary reticule cursor will show the point of action. For example, with **Snap To - Points** set the point that is nearest to the mouse cursor will be highlighted with a reticule cursor that is magenta in color. If we click with the left mouse button the mouse cursor will jump immediately onto that point. This makes it possible to exactly pick out the location of a given point, for example to edit its position.

Checking **Edit - Snap To - Lines** will cause the snap cursor to jump to the nearest coordinate defining a line. In the illustration above, as the regular cursor is brought near the end of a line the snap cursor will appear at the end of the line.

As the cursor is brought near the end of the other line, the snap cursor will jump to the end of that line.

**Snap Modes in Drawings:**

- **Graticule**
  - Snap to graticule intersections if View - Graticule is enabled.

- **Grid**
  - Snap to grid intersections, if View - Grid is enabled. Frequently used when adding controls to Forms.

- **Areas**
  - Snap to the coordinates that define areas.

- **Lines**
  - Snap to the coordinates that define lines.

- **Points**
  - Snap to points.

- **Segments**
  - Snap to arbitrary locations on segments defining line or area objects.

- **Selection**
  - Snap only to those objects in the selection. Requires snap to areas, lines or points to be enabled.

More than one snap mode can be set simultaneously. For example, if **Snap to Areas**, **Snap to Points** and **Snap to Selection** are all checked then the cursor will jump to the nearest point or area coordinate that is in the selection.

**Grid and Graticule Snap Modes**

Snapping to grids and graticules is supported in most components; with drawing, image, labels, map, profile, surface and theme windows support snapping to grids and graticules.
Snap Modes in Layouts:

- **Grid**: Snap to grid intersections, if View - Grid is enabled.
- **Elements**: Snap to the corners of an element.

Changing Snap Action with TAB and Space Bar

- **TAB**: Depressing the TAB key jumps the cursor immediately onto the snap reticule. This is used when we want the mouse cursor to jump immediately to the snap point without a mouse click.
- **Space Bar**: Depressing the space bar turns off snap temporarily until the space bar is pressed again. This is very useful to temporarily turn snap off. When snap is turned off using the space bar the status bar will not show snap modes enabled.

Example

Suppose we would like to use the Tracker tool to measure distances from one point to other points. With Snap to Points checked, we position the snap reticule over the desired Tracker origin point and click with the left mouse button. Now, we move the mouse cursor near the point for which tracker measurement is desired so that the snap reticule highlights that point and we press the TAB key: the mouse cursor will immediately snap onto that point and the Tracker readout will provide the distance to that point. To measure the distance to another point we can move the reticule onto another point and press TAB again and the mouse cursor will snap to that point, and so on.

Snap modes also work in the middle of other commands. For example, suppose we would like to draw a circle that is centered on one point and that has its edge exactly on a particular area.
We push in **Snap to Points** and **Snap to Areas** and then use the **Insert Circle on Center** drawing tool to click on the first point and then drag the circle out to the area. In the example above, the magenta snap cursor appears at the point initially clicked. As we bring the mouse cursor near the area a square secondary snap cursor will appear at various coordinates that define the area that are nearest to the mouse cursor. When we release the mouse drag the circle will be created as denoted by the magenta snap cursor and the secondary box cursor.

**Example**

In this example we will reposition labels to the intersections of a grid.

Consider an unbound labels component. This was created as a bound labels component of place names in Mexico using the Mexico example drawing as a parent and then unlinked to create an unbound labels component. The example would work the same using a bound labels component, but if we had used a bound component the parent objects would have been shown in schematic preview in the background and thus the display might not have been as clear.

The component has been projected into Lambert Conformal Conic projection using default **Suggest** button values.

We first turn on grids by using **View - Grid** to specify a grid for the view to display. We use default values for the grid.
Next, we select one of the labels to be moved for editing by CTRL-ALT clicking on it.

When the label is selected for editing an edit handle appears at the tie point for the label.

At this point we can turn on Snap to Grid, but when we do so we discover we cannot click on the label's edit handle to drag it to a different position. When we try to do so we find that the Snap mode moves the cursor reticule onto the nearest grid intersection and not onto the edit handle as we would like.

The solution is to tap the spacebar on our keyboard to temporarily turn off Snap mode. We can then click and hold onto the edit handle so the square edit cursor appears on the edit handle. We hold the click down...
...while we simultaneously tap the **spacebar** again to turn **Snap** mode back on. The edit cursor now under the influence of **Snap** mode immediately jumps to the grid intersection nearest to the mouse arrow cursor.

When we release our click-and-hold the label snaps to the grid intersection.

We can click on the label again (which we can do without using the **spacebar** because it is now at a grid intersection to which the cursor will jump in **Snap** mode) and, holding the click, move the mouse arrow cursor to a different location.
As the mouse arrow cursor nears the upper grid intersection the square edit cursor jumps to that intersection.

When we release the click-and-hold the label will snap to the upper intersection.

**Snap Tolerance**

The **Snap Tolerance** parameter may be optionally specified in the Tool Properties pane to control how close the mouse must be to a snap item before a snap mode takes effect.

When a snap mode is on, the **Snap Tolerance** setting gives the distance in screen pixels or in physical units within which the cursor must be to a given snap item (such as a line when snapping to lines) before the cursor snaps to that item.

When the cursor is outside the snap tolerance range, snaps are not effective. Setting a small snap tolerance is therefore a good way of retaining the benefits of snap modes without having the snap cursor jump about in a disconcerting way if the mouse is nowhere near a snappable item.

**Layer Snap Restrictions**

Using Layer Restrictions a map layer can be made non-snappable. When a layer is so restricted, any snap modes will ignore items in that layer.

**Note**

Snap must find the nearest point for every mouse motion. The snap cursor will lag behind mouse motion if there is a very large number of points on screen, if we move the mouse very rapidly (snap will catch up when we slow down), or if the system is heavily loaded with a background task.

**Troubleshooting**

If snap appears not to be working, try pressing the spacebar to see if snap has temporarily been turned off. When snap is on the status bar will show snap mode turned on. If snap has been turned off using the space bar, the status bar will not show snap mode enabled even if a snap mode toolbar button is pressed in.
See Also

Editing with Snap
Editing Forms
Autocomplete with ALT
Tool Properties
Edit - Shared Edit

The Edit - Shared Edit setting controls the operation of mouse-based editing commands. When Shared Edit is on, any click and drag of the mouse will affect all objects that have a defining coordinate at the mouse click location. When Shared Edit is off, only that object that has been selected for editing will be affected.

Shared Edit may be turned on and off using CTRL-E as a keyboard shortcut ("E" being a mnemonic for "Edit"), or by clicking the Shared Edit choice in the Edit menu.

When Shared Edit is on, a box will appear around the Shared Edit icon in the Edit menu.

When Shared Edit is off, there will be no box around the Shared Edit icon in the Edit menu.

Example

The easiest way to show the effect of Shared Edit is to turn it on and try a simple example.

Consider a drawing that has four lines. Three of the lines have been drawn using the Snap to Lines snap mode so that their ends are exactly at the end of the fourth line. At the ends of the four lines five points have been placed, also using Snap to Lines snap mode, so that the points are exactly at the ends of the lines.

This is a fairly typical example of lines and points that might occur in a drawing of, say, a system of roads or a network, where the ends of lines are exactly coincident and where points might be placed exactly at the ends of lines. Suppose we would like to edit this drawing to move, say, the position of one of the points and we would like to simultaneously change the position of any incident lines.

That would be tedious to do without Shared Edit capability, but with Shared Edit mode turned on it is very easy to do.
Suppose we would like to move the central point and to move all incident lines with it. We being by CTRL-ALT clicking on the point to select it for editing. An edit handle appears.

We can now click on the point and drag it to a new location.

With Shared Edit mode turned on, when the point moves, the coincident coordinates of all incident objects move with it. The four lines with ends exactly under the point will have their ends also moved. What is happening is that the coordinates defining the end of each line are also changed when the point's coordinates are changed by moving the point.

This works regardless of which object has been selected, and it even works if the different objects are in different layers and we are editing them in a map.
Suppose, for example, we **CTRL-ALT** click on the uppermost line to select it for editing. Edit handles appear at the two coordinates that define the line.

We can now click and drag the edit handle on the bottom coordinate to move it.

With **Shared Edit** mode on, when we move the edit handle for the line we change the line, but the point at that location also moves and so do the ends of the other three lines.
Let's consider what would have happened if we did not have **Shared Edit** mode on. Suppose we **CTRL-ALT** click the upper line to select it for editing.

Once again, we could click and drag the bottom edit handle to move it.

However, if **Shared Edit** mode is turned off then moving the bottom edit handle will move only the coordinate for that one object and will not move the point and the ends of the other three lines that were coincident with it.

**Editing Areas with Shared Edit**

**Shared Edit** mode is very useful for editing areas where areas are adjacent. It's often the case with geographic maps that areas will be drawn adjacent to each other. If we want to edit the boundary of an area to move it slightly we will also need to correspondingly move the boundary of any adjacent areas to avoid creating gaps or overlaps.

Geographic regions such as provinces or states are often represented as areas in geographic maps. The "border" between two states in an error-free geographic map is a place where the boundaries of two different areas coincide. Let us digress for a moment to make this concept clear:
The illustration above shows a map that has two layers, each containing an area object. The areas have been drawn so they are exactly adjacent to each other along a common border. What appears to be a single line, a "border," is in fact simply the place where the coordinates of two different area objects happen to coincide.

We can see this by clicking the layers on/off for display that contain the two areas.

One layer contains an area colored in gray. This area is defined with seven coordinates, five of which define the right edge of the area.

The other layer contains an area colored with a blue pattern. This area is also defined with seven coordinates. The left edge of the blue patterned area is also defined with five coordinate locations.

What gives the appearance of a common “border” is that the five coordinates defining the right edge of the gray area are exactly coincident with the coordinates defining the left edge of the blue area.

This is the usual case in most geographic maps. Unless there is a boundary dispute between two states, normally where the territory of one state ends is where the other state begins, so there should not be any gaps or overlaps at the place where two areas meet.

To move the common boundary line, we want to move the coordinates of both the gray area and the blue area simultaneously. To do this, we turn Shared Edit mode on and we will use regular editing commands to change the shape of one area with the other area's shape being changed automatically.
For example, we can select the area on the right for editing by **CTRL-ALT** clicking it so edit handles appear at each of the area’s coordinates and then we can drag edit handles about to change the shape of the area.

To move the central part of the border we can click and drag the leftmost edit handle to the right.

With **Shared Edit** turned on, the result is that the corresponding coordinate for the gray area is also moved to the right. This changes the shape of the gray area as well so that no gaps or overlaps appear. Note that this works even though the two areas are in different layers in the map window in which we are working.

Had we done this same edit with **Shared Edit** turned off, only the coordinate for the area selected for editing would have been moved. The coordinate for the gray area would not have been moved and thus a gap would appear between the two areas.
**Edit - Instant Data**

*Instant Data* make repetitive data entry a snap in drawing windows or in drawing layers in map windows. This is a mode command. It is either checked or not checked in the *Edit* menu. When checked, all subsequent object creation commands, like *Insert Point*, will call up the *Instant Data* dialog.

When an object is created, the *Instant Data* dialog allows us to enter a value into a specified field. The field box will be loaded with all write-enabled fields available in the drawing's table. Choose a desired field and enter the value. Thereafter, when *Instant Data* opens it will show that field and value by default. The *Instant Data* dialog remembers the last field used and opens with the field name pre-loaded and the cursor positioned in the value box. Type in the value desired and hit *Enter* or click on *OK*.

![Object Fields Dialog](image)

**To use Instant Data**

1. Open a drawing window or click on a drawing layer in a map to make it active.
2. Click *Instant Data* in the *Edit* menu to check it ON.
3. Choose a tool to insert the objects you want, for example, *Insert Point*.
4. Click the location where you wish to insert a point. The *Instant Data* dialog will open.
5. In the *Column* box, choose the field for which you want to add data.
6. In the *Value* box, enter a value for that field.
7. Press *OK*.
8. Continue adding objects. Each time the *Instant Data* dialog will appear preloaded with the name of the field and the last value added. Change the values as desired. If for some objects you wish to add a value in a different field, choose the new field in the *Column* box.
9. When finished using *Instant Data*, click it off in the *Edit* menu.

Use *Instant Data* when you must create many objects and wish to add a value for each. For example, suppose we need to add many points to a map and we wish to enter a name for each in a field called *Name*. Check *Edit - Instant Data* and then we can use *Insert Point* to click where we want each point. Enter the name, hit *Enter* and click on the next point.

*Instant Data* thus allows a very rapid repetitive data-entry rhythm: click a point - enter a value, click a point - enter a value, and so on. Please note that because *Instant Data* is used to load values into a database field for newly-created objects the *Instant Data* dialog only appears when there are fields defined in the drawing's table. If we create a new, blank drawing the *Instant Data* dialog will not be available until we add some fields to the drawing's table.

*Instant Data* remembers the last value used for each drawing. If we turn on *Instant Data* and then add points to different drawing (say, by clicking into different drawing windows or by changing the active layer in a map), the *Instant Data* dialog will be loaded with the last field used and last value used for each drawing.
Using Instant Data when Several Objects are Created

Manifold commands can be used to create many objects at once. For example, if the Insert Areas tool is used with Create Areas, Create Lines, and Create Points modes on the area with a line object at the boundary and point objects at the coordinates that define the area.

When several objects at once are created, the instant data dialog tries to identify the "main" object for which a data field value is to be assigned, using the order of precedence: areas, lines and finally points. For example, when creating an area with a boundary line and points at the defining coordinates the area will be taken as the "main" object and the instant data dialog will set a data field value for the area.

If no object can be obviously identified as the "main" object the instant data dialog will not launch even if Edit - Instant Data is checked. For example, we can use the Insert Box tool with Insert Points mode set ON to create four points at the corners of a box. There is nothing to distinguish any of the four points as being more "main" than any of the other points, so the instant data dialog will not launch in this case.

Example

Suppose we have a map that shows various regions in a drawing. In another drawing we would like to show the location of sampling stations as points. Each station has a name.

1. Create a new drawing and drag it into the map.
2. Click open the table for the new drawing.
3. Add a text field called Name to the table.
4. Click on the map and click on the new drawing's tab to make it the active layer.
5. Check Edit - Instant Data
6. Choose Insert Point
7. Click at a location where a sampling station is to be located. A point will be created and the Instant Data dialog will appear.
8. Choose the Name field in the Field box and then enter the text for the name of this sampling station in the Value box. Press Enter or click OK.
9. Click at the location of the next sampling station. The Instant Data dialog will appear once more with the Name field already loaded and the most-recently entered text in the Value box (this facilitates entering the same value over and over). Enter the name for this next sampling station.
10. Repeat the cycle above until all the sampling station points have been created.

When finished using Instant Data, take a moment to uncheck Edit - Instant Data

See Also

See the Add Points with Instant Data example topic for an example of adding points, using Instant Data and moving points by editing intrinsic fields.
**Edit - Template**

The Template menu command appears in the Edit menu when a print layout is open and has the focus. This command allows application of layout templates or other layouts to the current layout, or saving the current layout to a file for use as a template.

- **Apply**
  - **Apply a template** - Apply a built-in template to this layout.
  - **Apply a layout** - Apply a different layout in the project as a template to this layout.

- **Apply File**
  - Apply a layout template stored in a file to this layout.

- **Save**
  - Save this layout to a file for future use as a layout template.

**See Also**

- Layouts
- Layout Templates
- Custom Layout Templates

**View Menu**

**View - Columns**

Show or hide columns in tables and specify their order. The View - Columns dialog shows all fields available in the table, with a readout at the bottom of the dialog that shows how many columns are currently shown out of the total number or columns in the table. The dialog also shows the data type of each column's field.

For tables linked to drawings, the View - Columns dialog will also list all intrinsic fields available. Intrinsic fields are system-generated fields that report information, such as latitude and longitude location, for each object.

Check the column box to show that column. Click on a column to highlight it and then the Move buttons to move it up or down in order.

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contact Name</td>
<td>Text (Uni)</td>
</tr>
<tr>
<td>Contact Title</td>
<td>Text (Uni)</td>
</tr>
<tr>
<td>Address</td>
<td>Text (Uni)</td>
</tr>
</tbody>
</table>

- **Show All** - Show all columns in the table.
- **Hide All** - Hide all columns in the table.
- **Show Inverse** - Uncheck all checked columns and check all unchecked columns. A fast way to show all but one column: click Hide All, check the one column not desired and then click Show Inverse.
- **Move to Top** - Move highlighted column to the top of the list. This places the column at the leftmost position of the table.
- **Move Up** - Move highlighted column up one position in the list. This moves the column one step to the left in the table.
Move Down - Move highlighted column down one position in the list. This moves the column one step to the right in the table.

Move to Bottom - Move highlighted column to the bottom of the list. This places the column at the rightmost position of the table.

See the main Tables topic for details on tables.
View - Selection Filter
Enabled for tables. Push in to filter the table so that only selected records are shown

With the focus on a table window, push in the Selection Filter button either in the main toolbar or in the View - Filter dialog. The table will then show only selected records. Records will appear and disappear from a table window filtered Selection Filter as they are added to or subtracted from the Selection.

Example

Import the sample drawing of Mexico provinces from the Manifold CD. Open the drawing and open the drawing's table. Click on the table window so it has the focus and push in the Selection Filter button.

Initially, there is no selection in the drawing.

The table therefore shows no records since there is nothing in the selection and with the Selection Filter button in the table will show only selected records.
If we click on the province of Durango to select it…

...It will immediately appear in the table as well.

If we use Select Add mode and use Select Touch to click on two more provinces…
Components, Dialogs and Controls Reference

...Their records also will immediately appear in the table.

**Query Toolbar Restricted to the Selection Filter**

When the selection filter button is engaged subsequent operations of the query toolbar will apply only to those records that are shown in view by the selection filter. This makes it easy to run simple queries that refine the result of a previous query toolbar operation.

Suppose we select records with values in column A greater than 0. Suppose we now want to select records with values in column B greater than 0, regardless of what is in column A. That's easy to do as we simply run the query toolbar again.

Now suppose we again select records with values in column A greater than 0, but this time we also want to find within those selected records only those records that have values in B greater than 0 as well. After making the selection with the query toolbar to find those records with values in A greater than 0 we turn on the selection filter so that those records are shown and then next we run the query toolbar one more time to find those records that also have values in B greater than 0.

**See Also**

**Selection**
View - Sort

This dialog sorts tables using multilevel sorts on several fields at once. It shows all the columns in the table and allows specification of ascending or descending Order for each field. Specify the order of the column by double-clicking into the Order field and choosing either ascending or descending order.

Manifold reads the precedence to be used from the top down to the bottom of fields listed in the Sort dialog. Fields that do not have an ascending or descending sort order specified will be ignored.

- **Move to Top** - Move the highlighted filter rule to the top of the list.
- **Move Up** - Move the highlighted filter rule up one position in the list.
- **Move Down** - Move the highlighted filter rule down one position in the list.
- **Move to Bottom** - Move the highlighted filter rule to the bottom of the order of the list.
- **Use Case** - Specify case sensitive or case insensitive (the default) sort. A case sensitive sort will place records beginning with capital letters at the top of the table and those beginning with the same letter in lower case at the bottom of the table.
- **Use Side Whitespace** - Use or ignore leading and trailing spaces in sort order. Suppose a red dash "-" indicates a space character: If turned on, "A" sorts before "B"
- **Use Interior Whitespace** - Use or ignore interior spaces in sort order. Suppose a dash "-" indicates a space character: If turned on, "A-B" (with two spaces between "A" and "B") sorts before "A-C" (with one space between the letters).

**Order** Sort this field by ascending or descending order. Ascending order means records beginning with A will be at the top of the table and those beginning with Z will be at the bottom. Descending order will have records beginning with Z at the top of the table.

To Sort a Table using Multilevel Sort

1. In the View - Sort dialog, click on the field to be used as the topmost level sort.
2. Use Move Up to move it above any other field used for sorting.
3. Click into the Order cell for this field and specify ascending or descending sort.
4. Repeat the above process for fields to be used for sub-sorting.

Example

We have a database we wish to sort first by State and then by City fields.

<table>
<thead>
<tr>
<th>State</th>
<th>Order</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

 ascendant, descendent
Double-click into the **Order** cell for the **State** column and choose *ascending*.

<table>
<thead>
<tr>
<th>Column</th>
<th>Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>City</td>
<td></td>
</tr>
<tr>
<td>State</td>
<td>ascending</td>
</tr>
</tbody>
</table>

Double-click into the **Order** cell for the **City** column and choose *ascending*.

<table>
<thead>
<tr>
<th>Column</th>
<th>Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>State</td>
<td>ascending</td>
</tr>
<tr>
<td>City</td>
<td>ascending</td>
</tr>
</tbody>
</table>

Click on the **State** column to highlight it and use **Move Up** to move it above the **City** field in the list if it is not already above it.

This will result in a table that has all the records sorted by state first and then within each state they will be sorted by city names. To find a particular record we can scroll down to the state and then look within the state for the particular city name.

<table>
<thead>
<tr>
<th>Column</th>
<th>Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>City</td>
<td>ascending</td>
</tr>
<tr>
<td>State</td>
<td>ascending</td>
</tr>
</tbody>
</table>

If we were to use **Move Up** to move the **City** field above the **State** field the resultant sort would first sort records by City and then by state.
We could then scroll through the database and find, say, 19 instances of the city named **Washington** in the US with each record further sorted by state, so that table would appear for those records as:
**View - Zoom**

Menu commands analogous to zoom button commands.

- **Zoom In** - Magnify the view as if seen from a closer distance.
- **Zoom Out** - Reduce the view as if seen from farther away.
- **Zoom To Fit** - Zoom so that the component fits within the current window size.
- **Zoom To** - Zoom to a given zoom level, expressed as a specific scale if the component is projected.

When zooming with images, zoom in and zoom out will zoom to discrete zoom levels. To zoom to a particular level, use the **View - Zoom To...** menu command. The **Zoom To** menu command allows zooming to a particular scale when components are projected.

Images and surfaces will include a **Native** option in the **Zoom To** choices that will zoom the image so that one image pixel is one screen pixel. The **Native** option with drawings is the only **Zoom To** choice available in unprojected drawings. Project the drawing into some convenient projection for choices of specific scales.

See also the Zoom combo in the navigation toolbar to rapidly zoom to some specific zoom scale.

When zooming, each zoom step will be made using the **Zoom magnification factor** specified in the Tools - Options - User Interface dialog. This factor specifies the change in magnification to use when zooming in or out in the range from 1.1 to 4.0. The default value is 2.0.

**Scroll Wheel**

When using a mouse with a scroll wheel we can use the scroll wheel to zoom in and out. This works even in the middle of commands such as the Edit Toolbar editing commands. By default the zoom will be centered on the mouse position. Holding the **CTRL** key while operating the scroll wheel will force the zoom to be near the center of the opened window.

**Comparison to Edit - GoTo**

To zoom to a particular selection, an address or many other options, use the Edit - GoTo command.

**The ALT Key and Zoom**

Pressing the **ALT** key when using either **Zoom In** or **Zoom Out** will toggle zoom function in/out. For example, holding down the **ALT** key when the **Zoom In** command will have the same effect as **Zoom Out**. This use of the **ALT** key allows us to choose either **Zoom In** or **Zoom Out** and then rapidly browse by clicking with the mouse with the right hand and using the **ALT** key with the left hand to zoom in or out as desired. [Or vice versa if you are left handed.]

**Note:** The **ALT** key works as described above only if Snap modes are not on.

**CTRL Zoom Shortcut**

Pressing the **CTRL** key and the * asterisk key on the numeric keypad at the same time will zoom an image to native zoom, with one image pixel equivalent to one screen pixel.

**Matching Scale**

In the Tools - Options dialog check the **Adjust display scale for monitor resolution** box. That will match the scale used in Manifold to what is literally shown on the monitor given monitor size, DPI and other factors.
When working with layouts we can then choose **View - Zoom to** and choose a zoom of 1:1 so that the layout is shown exactly real size. If we have a large enough monitor we can then see the layout exactly as it will be printed with, for example, an A4 sized paper layout being displayed as a full-sized A4 sheet of paper on the monitor.
The **View - Graticule** command displays or creates a grid of lines that are parallel to meridians of longitude and parallels of latitude. Graticule lines will be curved in geographic projections.

Graticules may be shown as system items in the window or may be saved as objects into a drawing. Graticules may also be used with Snap to precisely locate map clicks.

- **Show Graticule**: Check box to show graticule.
- **From / To**: The beginning and end Longitude and Latitude values the graticule should span.
- **Spacing**: Longitude and Latitude spacing of graticule lines.
- **(Units box)**: Not enabled for graticules, which always use degrees.
- **Same Spacing**: Check so that any change in Longitude spacing will update Latitude spacing as well and vice versa. Use if the same Longitude and Latitude graticule spacing is desired.
- **Segments**: The number of coordinates to use per graticule line. More segments will allow for smoother curves if lines created with graticules are projected.
- **Style**: Graphics style used to show graticule.
- **Size**: Width of line in printer's points to use for the graticule line.
- **Suggest**: Load suggested defaults for the current view.
- **Create**: Create actual line objects in the drawing for the grid. Use carefully, since every press of the Create button will create another set of lines.

Most "whole world" graticules are laid out from -180 to 180 degrees Longitude and from -70 to 70 degrees Latitude in spacing of 10 degrees. This provides a pleasing effect when used with maps of the entire Earth.

If a very large number of graticule cells (greater than 10000) would be created given the specified settings, Manifold will ask for a confirmation in case of error.

### Creating Graticules as Objects

The graticule that appears when the **Show Graticule** box is checked is not part of any layer. It is something created by the system and overlaid upon the map window. It does not really exist and cannot be selected or otherwise manipulated. The default graticule color is set by an option in Tools - Options.

We frequently will want to create actual objects, lines, in the form of the graticule to allow us to better manipulate the graticule in graphics arts effects. For example, we might wish to position the graticule in a drawing layer in a map so that it is above some layers (oceans objects) and below others (continents). Alternately, we might want to select some of the lines that make up the graticule and delete them so we have a graticule in some places but not in others. This is of interest sometimes to create a proper appearance when using "interrupted" projections.

To create line objects, set up the graticule as desired in the **View - Graticule** dialog and then press the Create button. The graticule will be created in the active drawing as line objects. These may then be manipulated like any other lines. In maps it is often convenient to first create a new blank drawing with a right click on a layer tab and using **Add - New Drawing**, to rename the new drawing "graticule" and to then create the graticule in that layer.

**Example**
If we create a graticule on a world map (seen in \textit{Latitude / Longitude} projection) it will overlay all objects in the map. The graticule shown extends from -180 to 180 degrees longitude and from -70 to 70 degrees latitude using a spacing of 10 degrees.

We can use \textbf{Add - New Drawing} to create a new drawing and then create a graticule by pressing the \textbf{Create} button in the \textit{View - Graticule} dialog.

To see the new set of graticule lines, we uncheck the \textbf{Show Graticule} box in the dialog (otherwise, the graticule shown by the system will precisely overlay and thus hide the new graticule just created). We can move the new drawing to a layer below the world map drawing and change the color of the lines in the graticule.

\textbf{The Difference between Graticules and Grids}

Graticules are always expressed in geographic coordinates (latitude and longitude) while grids are expressed in the native X and Y coordinates of the coordinate system of the component. For components using the \textit{Latitude / Longitude} “non-projection”, both graticules and grids will appear as a grid of horizontal and vertical straight lines.

In projected coordinate systems, graticules will be created as curved lines (if necessary) to parallel the curved form of meridians of longitude or parallels of latitude in the projection. Grids, however, will always appear as a grid of horizontal and vertical straight lines.

In the image above a drawing of a world map is shown in \textbf{Robinson} projection with a graticule in magenta and a grid in blue. Note that the graticule has curved meridians while the grid uses straight lines only.
**Tech Tip**

Interrupted projections are not continuous coordinate systems. They employ multiple conversion domains with blank space between the different lobes of the conversion domains, as seen in the illustration below of the Mollweide Interrupted projection.

![Mollweide Interrupted Projection Illustration](image)

It is absolutely essential to check the **Clip Coordinates** box in the **Projection** dialog whenever using such projections. This box causes Manifold to clip each object so that it exists only within the allowed conversion domain lobes and does not extend or cross through disallowed blank space. This is a highly computationally intensive process so the **Clip Coordinates** box is not checked by default, so that significant overhead is not imposed if it is unnecessary.

Dealing with the separate conversion domain lobes of an interrupted projection requires a manual approach to creating graticules, since the graticule lines normally extend through the blank space between lobes. Use the following procedure:

1. Create a latitude / longitude drawing.
2. Use the **View - Graticule** tool to create the desired graticule, using the option to **Create** the graticule as line objects.
3. Project the drawing into the desired interrupted projection, making sure to check **Clip Coordinates**.
4. Edit the graticule lines by selecting undesired lines and deleting them, or by adding lines.

For many uses the fastest method is to add lines. Let's consider an example of creating a graticule for use with **Mollweide Interrupted**. Suppose we begin with a latitude / longitude drawing in which a graticule was created with lines every 10 degrees from -180 to 180 longitude and from -70 to 70 latitude.

![Graticule Illustration](image)

After projection into Mollweide Interrupted with **Clip Coordinates** checked, the lines that appear on the edges of the conversion domain lobes will have been deleted.
We can add lines by clicking Snap To Lines and then using the Insert Line tool to add lines between the "dangling" parallels. This goes very rapidly with less than a minute required to complete the graticule.

The result will be a graticule grid with lines restored that were deleted by Clip Coordinates. This procedure was used to create the graticule seen in the interrupted projection illustration above.
View - Grid
Show a grid of horizontal and vertical lines that may be used as a guide when editing. Grids are often used with Snap to precisely locate mouse clicks. The default color for grids is set in Tools - Options.

Usage with drawings, images, surfaces, labels and maps

In drawings, images, surfaces, labels and maps the View - Grid dialog provides the following options:

- **Show Grid**: Check box to show grid.
- **From / To**: The beginning and end X and Y values the grid should span.
- **Spacing**: X and Y spacing of grid lines.
- **(Units box)**: Choose units to be used.
- **Same Spacing**: Check so that any change in X spacing will update Y spacing as well and vice versa. Use if the same X and Y grid spacing is desired.
- **Segments**: The number of coordinates to use per grid line. More segments will allow for smoother curves if lines created with grids are ever projected.
- **Style**: Graphics style used to show grid lines.
- **Size**: Width of line in printer's points to use for the grid line.
- **Suggest**: Load suggested defaults for the current view.
- **Create**: Create actual line objects in the drawing for the grid. Use carefully, since every press of the Create button will create another set of lines.

Default settings for the above options will be presented that cover the current view with a grid, using what are likely to be reasonable settings for stepping.

If a very large number of grid cells (greater than 10000) would be created given the specified settings, Manifold will ask for a confirmation in case of error.

Usage with forms and print layouts

In forms and print layouts, the Grid dialog provides a simpler set of controls, since the grid is automatically created over the entire layout or form:

- **Show Grid**: Check box to show grid.
- **Step by X**: X (horizontal) spacing of grid lines in the given units.
- **Step by Y**: Y (vertical) spacing of grid lines in the given units.
- **(Units box)**: Choose units to be used.

Creating Grids as Objects

The grid that appears when the Show Grid box is checked is not part of any layer. It is something created by the system and overlaid upon the map window. It does not really exist and cannot be selected or otherwise manipulated. The default grid color is set by an option in Tools - Options.

We frequently will want to create actual objects, lines, in the form of the grid to allow us to better manipulate the grid in graphics arts effects. For example, we might wish to position the grid in a drawing layer in a map so that it...
is above some layers (oceans objects) and below others (continents). Alternately, we might want to select some of the lines that make up the grid and delete them so we have a grid in some places but not in others. This is of interest sometimes to create a proper appearance when using “interrupted” projections.

To create line objects, set up the grid as desired in the View - Grid dialog and then press the Create button. The grid will be created in the active drawing as line objects. These may then be manipulated like any other lines. In maps it is often convenient to first create a new blank drawing with a right click on a layer tab and using Add - New Drawing, to rename the new drawing “grid” and to then create the grid in that layer.

Example

If we create a grid on a world map (seen in Latitude / Longitude projection) it will overlay all objects in the map. The grid shown extends from -180 to 180 degrees longitude and from -70 to 70 degrees latitude using a spacing of 10 degrees.

We can use Add - New Drawing to create a new drawing and then create a grid by pressing the Create button in the View - Grid dialog.

To see the new set of grid lines, we uncheck the Show Grid box in the dialog (otherwise, the grid shown by the system will precisely overlay and thus hide the new grid just created). We can move the new drawing to a layer below the world map drawing and change the color of the lines in the grid.

The Difference between Graticules and Grids

Graticules are always expressed in geographic coordinates (latitude and longitude) while grids are expressed in the native X and Y coordinates of the coordinate system of the component. For components using the Latitude / Longitude "non-projection", both graticules and grids will appear as a grid of horizontal and vertical straight lines.

In projected coordinate systems, graticules will be created as curved lines (if necessary) to parallel the curved form of meridians of longitude or parallels of latitude in the projection. Grids, however, will always appear as a grid of horizontal and vertical straight lines.
In the image above a drawing of a world map is shown in **Robinson** projection with a graticule in magenta and a grid in blue. Note that the graticule has curved meridians while the grid uses straight lines only.
**View - Legend**

A **Legend** is a small display that summarizes formats used in a drawing, surface, image, labels component or map. By default, a legend shows object samples in a column to the left with a text caption describing each object sample to the right. Legends are tied to the component for which they are created. Manifold will remember the legend settings for that component.

**See Also**

Adding Legends

**Legends**

Adding a Legend
View - North Arrow

Display an arrow pointing North using a choice from various compass arrow styles.

Show north arrow  Check to display north arrow.

Align  Positioning of the north arrow: Left Bottom, Left Top, None, Right Bottom or Right Top. If None, the north arrow can be dragged to different locations.

Style  Choose from various graphical styles.

Bearing  Rotate the arrow by the given number of degrees. Very unwise to do except in special situations.

Size  Size in points.

Colors  Foreground and background colors for the north arrow.

Font  Font control for N, S, E and W letters that appear in some styles. Use this button to change the font used, the size of the font, etc.

Halo Text  Surround text letters with a halo of background color. Useful to provide a contrasting aura of color that allows letters to stand out no matter if the map behind them is light or dark.

Once a north arrow has been added to a component window, we can right click on it and choose Properties from the context menu to change the above settings.

The Tools - Options - Font dialog allows setting the default font to be used for those styles that include letters.

Legends, north arrows and scale bars can be copied and pasted between components. Copying a particular style of north arrow from an existing component and pasting it into a new component can be significantly faster than configuring a new north arrow from scratch.

Bearing Option

Manifold components are always displayed "North up" based on the assumption that they have been created that way in a geographic context. All correctly georeferenced components will always be "North up" so for every correctly georeferenced component the North Arrow should use a Bearing of 0 degrees.

On rare occasions we may work with drawings or images that are not georeferenced, for example, such as a factory diagram imported from AutoCAD DXF format, and which have not been drawn with "North up." In such
cases and only in such cases if we know the correct orientation for the North Arrow, we can use the Bearing value to provide a North Arrow that is not aligned straight up. Instead, the North Arrow will point to North at whatever Bearing rotation has been specified (with the default 0 degrees pointing straight up).

Note that changing the Bearing value does NOT rotate the drawing or map or other component in which the North Arrow is displayed: it simply rotates the North Arrow. Therefore, use of values other than 0 for Bearing is always a mistake for components that are correctly georegistered.

**Adding a North Arrow to a Print Layout**

See the Legends, Scale Bars and North Arrows in Layouts topic.
View - Scale Bar
Display a scale bar that shows the current scale of the map.

---

Show scale bar  Check to display scale bar.

Align  Positioning of the scale bar: Left Bottom, Left Top, None, Right Bottom or Right Top. If None, the scale bar can be dragged to different locations, but the bar can end up out of sight after pan or zoom operations. Using positions other than None always positions the bar relative to the open window or print layout to guarantee it stays in sight.

Style  Choose from various graphical styles.

Colors  Foreground and background colors for the scale bar.

Font  Settings for font used with those styles that include text.

Halo Text  Surround text letters with a halo of background color. Useful to provide a contrasting aura of color that allows letters to stand out no matter if the map behind them is light or dark.

Display  Number of units in the given units of measure. For example, if we wanted the scale bar to show 100 kilometers we would enter 100 and choose kilometers as the unit of measure.

Unprojected maps will allow only angular measures such as Arc Minutes, Arc Seconds, Degrees and Radians as units of measure.

Projected maps will allow a wide variety of standard linear units of measurement (including numerous international units of measure).

at least  Make the scale bar no smaller than this size in points.

at most  Make the scale bar no larger than this size in points.

Change color on underflow or overflow  If the units of measure and required display would force the scale bar to be smaller than the at least setting or
larger than the at most setting, change the color of the bar to show that it is out of range.

Once a scale bar has been added to a component window, we can right click on it and choose Properties from the context menu to change the above settings.

Scale bars are only approximations when large geographic regions are shown. To minimize distortion, use a projection that minimizes scale distortion.

Legends, north arrows and scale bars can be copied and pasted between components. Copying a particular style of scale bar from an existing component and pasting it into a new component can be significantly faster than configuring a new scale bar from scratch.

**Overruns and Underruns**

If the units of measure and required display would force the scale bar to be smaller than the at least setting or larger than the at most setting the scale bar will prepend a “>” or “<” character to the scale value.

**Adding a Scale Bar to a Print Layout**

See the Legends, Scale Bars and North Arrows in Layouts topic.

**Troubleshooting**

If a scale bar does not change size when the Display number is changed or if it otherwise misbehaves, consider if the number of units to display would cause the scale bar to be less than the minimum size specified (an underflow) or larger than the maximum size specified (an overflow).
View - Structure
The View - Structure menu choice allows us to show the enclosing box of objects and other structural details visually in drawings and labels components.

Use with Drawings

Check options to display them in the drawing window.

- **Object Boxes**: Show the minimum enclosing rectangle for each object.
- **Object Centroids**: Show the coordinate centroid for each object.
- **Branch Boxes**: Show the outlines of the minimum enclosing rectangles that enclose different branches of multi-branched objects.
- **Branch Centroids**: Show the coordinate centroid for each branch of multi-branched objects.
- **Inflection Points**: Show the location of coordinates (inflection points) that define an object.
- **Segment Points**: Show the midpoint of line segments (the straight portions of lines between the coordinates that define lines) in lines and the borderlines of areas.

Examples in Drawings

We begin with four objects, a multi-branched area (containing branches that represent "holes"), a single-branched area (an ordinary triangle), a line and a point. The illustrations show the result of each Structure option:
Use with Labels Components

Check options to show the object box outlines and tie points for labels in a labels window.

Object Boxes  Show the outline of the rectangular box that encloses the label text.
Tie Points  Show the tie point to which points labels are anchored.

Tie Lines  Show the tie lines to which line labels are anchored.

The object boxes and tie points and tie lines appear only in a labels component window. They are used to see the alignment of labels text relative to tie points.

Suppose we have a text label that consists of several lines of text.

Open the book to the first page and begin scanning words from left to right until their meaning is understood.

If we enable Object Boxes we see the outline that encloses the text. [We use a rather lengthy text label in this example because it shows the object box better.]

Open the book to the first page and begin scanning words from left to right until their meaning is understood.

If we enable Tie Points we see the tie point to which the text is bound and relative to which it is positioned.

Open the book to the first page and begin scanning words from left to right until their meaning is understood.

See the Aligning Labels topic for additional illustrations of tie points and tie lines.

Align Controls

The Align controls in the Format toolbar adjust the position of the label object box relative to the tie point for the label.

Align Left

Center Horizontally

Align Right

Align Top

Center Vertically

Align Bottom
The left-center-right and top-center-bottom buttons can be combined, for example, to create a right-bottom alignment.

**Note**

The text in the multiline label is taken from the "How to Use This Book" section of Mr. Bunny's Guide to ActiveX, by Carlton Egremont III, ISBN 0-201-48536-2. One of the funniest programming texts ever written, this slim volume presents ActiveX concepts in the form of conversations between a cartoon rabbit and farmer.

**See Also**

Aligning Labels

**Drawings - Object Coordinates**

**Editing Objects**
View - Full Screen
Display main Manifold window using the entire monitor screen, eliminating the title bar and menu bar. Press **F11** to enter full screen mode. Press **F11** again to exit full screen mode.

If toolbars are hidden using **Tools - Customize** and a window is maximized the result is a "full screen" display.
View - Refresh Data

Enabled for linked components such as linked drawings or linked tables and queries so that the linked component or query can be refreshed from the originating table. When the geocoded table controlling a linked drawing changes, the linked drawing will not be updated automatically within an interactive session of Manifold. Instead, the Refresh Data command must be issued.

When linked components are used in a Manifold IMS session we can set the Refresh linked components... parameter to some automatic interval, so Refresh Data is not necessary in IMS work.

Invoking the View - Refresh Data command on a query will cause cascaded refreshes within all dependent queries and linked tables.

Example 1

We have a linked table T and a query Q that selects some data from T. We run Q and it displays the resulting data.

We then you go to some other program and modify T outside of Manifold. Re-computing Q with View - Refresh Data will now first refresh T and will then recompute Q on the updated data in T.

Example 2

We have a native table T, a query S that selects some data from T and a query Q that selects some data from S. We run Q (in turn running S) and it displays the resulting data.

We then modify T so S returns a different set of records. Re-computing Q with View - Refresh Data will first re-compute S and then re-compute Q on updated data from S.
View - Refresh / Autorefresh View
Sets the redisplay behavior of windows and dialogs in Manifold.

**Autorefresh View**  
Refresh all windows on any change.

**Refresh**  
Used to manually refresh windows if Refresh Auto is OFF.

---

When working with very large maps and performing a series of steps, we might not be interested in viewing intermediate results if the price we pay is waiting for all windows to refresh with each step. In that case we can turn **Autorefresh View OFF** and then press **Refresh** once at the end of the procedure.

For example, when selecting objects using the query toolbar in very large drawings it can take a longer time than we would like for the system to redisplay the drawing using red selection color for each selected object. If we are selecting objects with the query toolbar just to delete them, we can save time by switching **Autorefresh View off**. We can then make a selection and delete the objects without waiting for the system to redisplay the drawing.

---

**Tech Tip**

While turning **Autorefresh View off** is very useful in some cases, be careful to turn this option back on when the tasks calling for suppression of autorefresh are finished. Otherwise, it is very easy to forget to turn it back on and later wonder why selections, etc., are not appearing as expected.
View - Display Options

The View - Display Options dialog specifies parameters used to control the appearance of components. When used with labels, the dialog allows us to specify how labels will be displayed or clipped from display. With surfaces, the dialog controls how the surface will appear when opened in a surface window or used as a layer in a map. With images, the dialog allows us to choose which channels of a multi-channel compressed image will be used for R, G, B and Alpha channels.

If the optional Surface Tools package is installed the dialog will also be available to control display options for profiles and elevations. See the Profiles and Elevations topic for more information.

Surface Display Options

By default, surfaces will appear as shaded surface renderings with the color of each pixel corresponding to the data value at that spot as adjusted for highlights and shadows from a Sun in the East. Other options include display of aspect or slope, or use of shading. See the Surface Display Options topic for examples, including details on using fixed interval palettes and customization of palettes. Palette controls are similar to those used in Thematic Formatting of drawings.

**Display**

Choose the computation used to display the surface in 2D.

- **Height** - Color each surface pixel by its height.
- **Aspect** - Color each surface by its aspect, that is, the azimuthal direction in which a tangent plane faces at that spot, for example, a north-facing section of hillside.
- **Slope** - Color each pixel by the inclination of the surface at that point. Slopes are computed using a weighted two-directional slope algorithm. Slopes are given in degrees.

**Palette**

Preset color combinations that may be applied to surfaces. Press **Apply** to apply the preset to the surface. Presets use a fixed number of intervals and corresponding colors.

- **Apply** - Apply the chosen palette preset to the Colors pane. This allows scrolling through the presets without changing colors until we press **Apply**. Pressing **Apply** only changes the color scheme in use for values. It does not change the formatting of the surface until the **OK** button is pressed.
- **Reverse** - Reverse the formats used in the colors box from high to low.
- **Interpolate** - Change the colors or sizes used in the colors boxes by interpolating between the top and the bottom boxes. A quick way of creating smooth gradients of colors.
- **Lighten** - Lighten all colors. Each click on the Lbutton lightens the colors a bit more.
- **Darken** - Darken all colors. Each click on the button darkens the colors a bit more.
- **Grayscale** - Convert all colors to grayscale.

- **Load from File** - Load a previously saved theme from an XML file. Works with all field types except lookup fields. See the Custom Palettes and Themes topic for information on how the XML file is structured.
- **Save to File** - Save this theme to an XML file. Works with all field types except lookup fields. See the Custom Palettes and Themes topic for information on how the XML file is structured.
**Shading** Shade the surface with highlights and shadows as if illuminated by the Sun from a given azimuth and elevation.

**Autocontrast** Automatically adjust contrast for even dispersion of light and dark tones.

**Azimuth** The angle to the Sun's position in degrees from North: North = 0, East = 90, South = 180 and West = 270 degrees. Keep in mind that in the Northern Hemisphere the Sun traverses azimuthal angles from East to South to West, but never North. Used to compute shading.

**Altitude** The Sun's altitude above the horizon in degrees. 0 is at sunrise or sunset and 90 is directly overhead. Used to compute shading.

---

**Display Options for Compressed Images**

The display options dialog appears when a compressed image window is open. Display options are not used for other types of images.

Compressed images are displayed as though they were RGBA images assigning channels from the compressed image data to R, G, B and Alpha channels.

(\text{Channel information}) The number of channels available in the image.

- **Red** Image channel to be used for red color.
- **Green** Image channel to be used for green color.
- **Blue** Image channel to be used for blue color.
- **Alpha** Image channel (if any) to be used for alpha channel transparency.

For an interesting effect, open the display options dialog for a compressed image and assign one of the image's channels to be used for the Alpha channel.

For more information on image display channels, see the Images and Channels topic as well as the RGBA Pixel Transparency topic.

---

**Display Options for Elevations**

Elevations are optional components that are available when the Surface Tools package is installed. See the Profiles and Elevations topic for more information.

- **Shade background** Color the background of the elevation using the same palette (if any) used to color the parent surface. On by default.

- **Display coordinates** Display dots at the location of each coordinate that defines the line used as a profile.

- **Include zero height** Show the entire range of elevations for the surface, from zero to the highest in the elevation display. Off by default, so that the full range of the elevation display is used to show the heights within the elevation only.

- **X axis (distances)**
Label coordinates Draw vertical lines and label the X axis at each coordinate that defines the profile line. Off by default.

Label even values Draw vertical lines and label the X axis at even values of distance. On by default.

Decimal digits The number of digits past the decimal point to show in labels on the X axis. The default value of 0 results in no digits past the decimal point.

Y axis (heights)

Label coordinates Draw horizontal lines and label the Y axis at each coordinate that defines the line used as a profile. Off by default.

Label minimum / maximum Draw horizontal lines and label the Y axis at the minimum and maximum values of height that occur in the profile line. Off by default.

Label surface breaks Draw horizontal lines and label the Y axis at the numeric values that define interval breaks in the palette used to color the surface. Off by default.

Label even values Draw horizontal lines and label the Y axis at even values of height. On by default.

Decimal digits The number of digits past the decimal point to show in labels on the Y axis. The default value of 0 results in no digits past the decimal point.

Display Options for Profiles

Profiles are optional components that are available when the Surface Tools package is installed. See the Profiles and Elevations topic for more information.

Display coordinates Display dots at the location of each coordinate that defines the line used as a profile.

Decimal digits in heights The number of digits past the decimal point to show in the Info pane for average, minimum and maximum heights when the focus is on this profile.

Labels Display Options

The View - Display Options dialog may be launched when a labels component has the focus. The dialog allows us to specify how labels will be displayed or clipped from display. The unit of measurement employed is the printer's point unit used in printing.

Point Labels Point labels are those used for points and for areas.

Offset X … points Distance in points in X (horizontal) displacement from the anchor point of the label. Increase distance to position labels farther away from the anchor point and to accentuate the effect of label alignment settings. 0 by default. Not used if alignment buttons are set to place the label directly upon the point.

Offset Y … points Distance in points in Y (vertical) displacement from the anchor point of the label. Increase distance to position labels farther away from the anchor point and to accentuate the effect of label alignment settings. 0 by default. Not used if alignment buttons are set to place the label directly upon the point.
Callouts Draw a line from the label to each point from which the label is generated.

Minimum length When a callout line is used, do not show it unless it is at least this long.

Optimize horizontal alignment Alter the horizontal alignment of labels to improve the positioning of labels.

Optimize vertical alignment Alter the vertical alignment of labels to improve the positioning of labels.

Line Labels Line labels are those used for lines.

Offset … points Distance in points that a line label should be offset from a line. 0 points by default.

Character Spacing Extra distance between characters in the line label text.

Label each branch If checked (default), will place a line label on each branch of a multi-branched line. If not checked, will place a line label only on the longest branch.

Multiple labels per branch If a particular line's branch is long enough, label it with multiple, repeating labels. This is useful for long lines.

Spacing The minimum spacing between multiple labels on the same branch of a line.

Orient labels left-to-right If checked (default), Manifold orients each label bound to a line to read from left to right as shown on the screen. When the option is turned off, Manifold orients each label so it flows from the beginning of the line to the end of the line.

Resolve overlaps between labels If checked (default), automatically hide some labels so that labels do not overlap.

Spacing Space allowed between labels in points before overlap resolution, when enabled, will take effect. Default is 1 point.

See the Label Display Options topic for details and examples.

Panes

View - Panes

Panes are sub-areas of main windows. They are often used within a main window to keep some information in view while another part of a window shows a changing display.

Manifold has several panes that can either be docked into the main Manifold window or that may be left undocked to float about “always on top” as independent dialogs. Panes may be opened and closed within the View - Panes menu. Panes may also be closed by clicking the X box in their caption bars.

Call Stack Display and manage the scripting engine's function call stack. Used to debug scripts.

Control Points Control points are used georegister images and drawings. The Control Points pane provides a listing of control points defined in the current image or drawing.

Errors Reports errors arising from compilations of .NET language scripts.

GPS Console Provides interactive control over a GPS (Global Positioning System) device connected to the computer.
via a serial port, including moving map functions and fetching information from the GPS.

**History** Displays a running history of actions and results.

**Info** Reports useful information about the active layer or window.

**Layers** Displays the layer stack in a map and allows specification of layer restrictions. The layers pane is more convenient to use with many layers than the tab controls at the bottom of the map.

When used with image windows, the layers pane turns individual R, G, B or alpha channels on and off and also controls whether the image border or background is displayed.

When used with print layouts the layers pane shows the vertical ordering of elements in the layout.

**Notes** A text dialog box allowing user entry of text notes and commentary. As the focus changes from one component window to the other, the contents of the Notes pane will change to the notes made for that component.

**Project** Shows all components in our project. It also shows how components are associated, such as a drawing and its table. We can use the project pane to open components in windows, to drag and drop components into maps, to rename components and to organize components within folders. The status area at the bottom of the pane will provide summary information about any highlighted component.

**Review** Interactive dialog used to resolve conflicts arising from multi-user editing.

**Selections** Provides a scratchpad for saving selections that have been made so that they may later be re-used.

**Tool Properties** Displays settings for the current tool in use, for example, threshold for Select Touch when selecting pixels within images or the opacity of a paintbrush when editing images.

**Variables** Display and manipulate variable values within scripts. Used to debug scripts.

**ViewBots** ViewBots are one-line analytic instruments that report a result or otherwise make a calculation using the contents of a window.

**Views** Saves specific views into a window at a given location and zoom level under a name. At any time we can jump directly to that view by clicking on its name.

**Watches** Watches are one-line dynamic instruments that watch and report upon values of variables in scripts or other process parameters. They are used to debug scripts by signaling watched-for conditions.

**World** The world pane allows setting linked views and position reticules and also shows the current location of the active window on a small world map.

Most users will leave one or more panes open while working with maps and tables. A typical setup would leave Selections, Info and ViewBots panes open on one side of the display with a Layers pane docked to the other side of the display. (This assumes we have a reasonably modern computer system with a larger screen and high-resolution display).

Panes will automatically become enabled or disabled as the focus switches between different windows. For example, the Layers pane is active only when a map window, image window or drawing window has the focus.
As the focus switches between different windows, panes will automatically be loaded with the correct information for that window. For example, the Selections pane will change to show the correct saved selections for whatever window has the focus.

Docking and Undocking

Panes can be docked into the main Manifold Window. Drag and drop a pane onto one of the main Manifold window’s borders to dock it. Panes may be docked into the upper, lower, left or right borders of the main Manifold window.

To undock a docked pane, double click on its margin. It will undock itself.

When a pane is dragged near to or on top of a margin using a click and drag, it will automatically dock itself. To override this, use a CTRL click and drag. This will allow us to drag a pane over top of a toolbar row or near to a window margin without the pane docking itself.

Manifold Keyboard Shortcuts for Panes

In addition to the standard Windows keyboard shortcuts, Manifold provides a set of keyboard shortcuts for opening/closing panes.

ALT-SHIFT-(letter) Show or hide a pane. Press the keyboard combination at any time to open or close a particular pane. The letter used is usually a mnemonic for the first letter of the name of the pane as shown below.

K Call Stack
C Control points.
E Errors
G GPS Console
H History
I Info
L Layers
N Notes
P Project
R Review
S Selections
T Tool Properties
A Variables
B ViewBots
V Views
W Watches
O World

ALT-SHIFT keyboard shortcuts are very important for turning panes off and on when working with systems that have small screens. If your screen display area is limited in size it will be inconvenient to open panes and leave them open because they will cover up too much of the active window. ALT-SHIFT shortcuts will allow you to instantly pop open a pane when needed and to make it vanish when it is not needed.
These shortcuts are listed in the View - Panes menu next to the pane they control.

**Autohide / Popup Panes**

Panes used for controls can be minimized when docked so that they autohide themselves when not in use and automatically pop open when the mouse cursor moves over their tab. Virtually every experienced Manifold user employs this feature to maximize the amount of screen area available for working with component windows.

Consider a Manifold workspace open with the project pane docked in its usual position.

Clicking on the Minimize control for the project pane will minimize the project pane into a small tab on the right-hand margin.
This frees up a lot of screen real estate for use with windows.

Moving the mouse cursor over on top of the project pane's tab (no need to click the tab)...
...will expand the project pane.

As long as the mouse cursor stays over the project pane it will remain expanded. As soon as the mouse cursor moves away from the project pane the pane will collapse again to a small tab.

Using autohide panes is a great way to maximise the available working area on a computer monitor. Consider, for example, the above view in which the project pane has collapsed into a small tab. In this view the active window has been maximized so that it uses all available space.
When open the project pane consumes much area which would be useful for work in the main component window. Although it is true that this example uses artificially small Manifold sessions (sized to fit reasonably as illustrations), even in the case of large, high-resolution monitors it makes sense to save work area if panes are not immediately needed.

**Ending Autohide / Popup Mode**

To end auto hide mode simply click on the pane's tab or click on the pane's caption. This will restore the pane to full size and will end autohide mode. In addition, although Manifold will remember autohide settings for future sessions, opening a new project will cancel autohide mode for the project pane. This allows the project pane to be immediately available for component usage as a new project is opened.

**Notes**

These illustrations show only the project pane. Most experienced Manifold users will have several panes open when working with projects. In such cases autohide capability is even more useful, because it enables multiple panes to be kept available as minimized tabs. When needed, each pane can expand to fit the entire space available for panes with a simple mouseover.

**See Also**

Panes may be rapidly turned on and off using the Tools - Customize dialog, which is especially useful when many panes at once must be turned on or off.

**Historical Note**

Manifold's autohide / popup panes capability originated in the July, 2007 User Meeting held in London, hosted by University College London (UCL) and attended by user community representatives from the US and Europe as well as by manifold.net factory representatives.

Dr. Muki Hakley of UCL presented a session on Usability Issues in GIS, in which he emphasized the importance of maximising available screen "real estate" for active workspace and not consuming a high percentage of the screen on displays of possibly infrequently-used controls. The presentation was based on the results of UCL research using sophisticated equipment to track the eye motions of users operating different GIS packages. Manifold scored much better in his reckoning than older GIS packages, but there was nonetheless room for improvement.

For the sheer fun and power of it, the User Meeting participants agreed to choose "something cool" to implement as a joint decision in Release 8, even though the release was only a few short weeks from publication. Dr.
Hakley's presentation was so well received and so persuasive that the consensus was clearly for implementing popup panes. And so they were implemented in the final stages of the Release 8 beta campaign.
View - Panes - Call Stack

The **Call Stack** pane is enabled when the Manifold Debugger is installed.

The Call Stack pane is used to display and manage the scripting engine's function call stack. It is used to debug scripts under the control of the debugger. Each row in the Call Stack pane shows a call.

- **Go To** - Enabled when a call within the Call Stack pane is highlighted. Go to that call. Double-clicking a line within the Call Stack pane is a shortcut to **Go To** that call.

- **Show Module** - Press in to show module (script) names for each call.

- **Show Line** - Press in to show the line number for that call within the module.

- **Show Line Position** - Press in to show the position within the line for that call. Useful when there is more than one call on the same line.

- **Show Language** - Press in to show the language used for each module.

For example, suppose we've stopped execution in a script called **A** at a breakpoint where another script is run.

```vbscript
Sub Main
    Set components = Application.ActiveDocument.ComponentSet
    Set anotherScript = components(components.ItemByName("B"))
    anotherScript.Run
End Sub
```

With all of the **Show** buttons pressed in, the Call Stack pane will report the following values:

If we click on the single call line shown within the Call Stack pane we can enable the **Go To** button. This is not particularly useful in such a simple example (since we are already at that call) but when the Call Stack pane shows many calls we can click on any of them to highlight it and to jump execution to that call.

**See Also**

See the Debugger topic for an example of using the Call Stack pane. Other panes used with the debugger include:

- Variables
- Watches
View - Panes - Control Points

Use the Control Points pane to add new control points to a drawing, image or map. Control points are used to georegister an image or drawing to the same geographic location and projection used by a "known good" image or drawing.

<table>
<thead>
<tr>
<th>Name</th>
<th>X / Longitude</th>
<th>Y / Latitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controll</td>
<td>-122.480997</td>
<td>37.811238</td>
</tr>
<tr>
<td>Control2</td>
<td>-122.048519</td>
<td>37.467911</td>
</tr>
</tbody>
</table>

The Control Points pane always shows a list of control points defined for the active image or drawing together with their coordinates. When we click on a different window the Control Points pane will switch context to show any control points defined for the active window.

The Control Points pane is also used to rename control points or to change their location by changing their coordinates. We can click into the Name field to change the name from the defaults, if desired. To change the coordinates of a control point we click into their X / Longitude or Y / Latitude fields.

- **New Control Point** - Add a control point by clicking with the mouse in the component window. A new control point will be added in the Control Points pane.
- **New Blank Control Point** - Add a blank control point row in the Control Points pane.
- **Delete Control Point** - Delete the highlighted control point. More than one control point may be highlighted by clicking on a starting point and then SHIFT-clicking on the ending point to highlight the starting and ending point and all points in between. All the highlighted points may then be deleted.
- **Register** - Georegister this drawing or image using matching control points in a reference component.
- **Load Points** - Load control points from points that exist in the drawing (enabled for drawings only).
- **Save Points** - Save control points as points in the drawing (enabled for drawings only).
- **Show Names** - Show the names of control points in the component.
- **Preview** - Show control points in the component.

**Name**

Name of the control point. Control points are matched between two components by using their names.

**X / Longitude**

The X or Longitude position of the control point.

**Y / Latitude**

The Y or Latitude position of the control point.

Adding a Control Point to a Component using the Mouse

1. Open the component. Zoom into the location where the control point is to be located if greater accuracy is required.
2. Click on the New Control Point button in the Control Points pane's toolbar.
3. Click the location in the component where you want the control point. It will appear in the drawing and a new control point row with coordinates will appear in the Control Points pane.
4. If you don’t like the name assigned to the control point by default double click onto the name to change it.

No matter what other tool modes are selected (for example, Zoom In or Center), after clicking New Control Point the mouse cursor will temporarily shift modes so that the next click locates the control point. If the mouse is clicked within the component window a new control point will be created. If the mouse is clicked anywhere else the New Control Point mode is abandoned and the mouse cursor will return to whatever state it was in before the New Control Point button was clicked. This is why we zoom in first to the region of interest where we wish to place the control point.

At times we will add new control points manually by specifying their coordinates.

**Adding a Control Point to a Component Manually**

1. Open the component to which the control point is to be added.
2. Click on the New Blank Control Point button in the Control Points pane's toolbar.
3. A new control point row appears with zero coordinates.
4. Double click into the X / Longitude and Y / Latitude coordinate fields to specify the desired coordinates.

**Show Names**

A control point appears by default with a label showing its name. The screen shots above show how control points appear in drawing layers at left and in images at right.

Clicking OFF the Show Names button removes the name labels from the display.

**Georegistration Dialog**

To georegister the active image or drawing, click on the Register button in the Control Points pane.

**Note:** Some options will be enabled only for certain methods or when sufficient control points have been defined.

<table>
<thead>
<tr>
<th>Reference</th>
<th>The name of the drawing or image to be used as a source of control points to be used as reference points.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td>One of the following algorithmic methods:</td>
</tr>
<tr>
<td></td>
<td><strong>Affine (triangulation)</strong> - Also called geometric transforms, affine methods can georegister a target component to a reference component using fewer control points than required by the Numeric method. However, affine registration requires care in placement of control points.</td>
</tr>
<tr>
<td></td>
<td><strong>Affine (scale, shift, rotate)</strong> - A fast method that is algorithmically equivalent to using Numeric with an Order of 1. This is the default method since it works with any number of control points.</td>
</tr>
<tr>
<td></td>
<td><strong>Simple (scale, shift)</strong> - Match components using XY translation and re-scaling only. Works with any number</td>
</tr>
</tbody>
</table>
of control points.

**Numeric (polynomial)** - Numeric matching uses numerical computation methods to transform one component to match another. A large number of control points are required for good matching but the method is algorithmically simple and fast. Available only when eight or more control points have been defined, thus allowing an order of \( 2 \) or greater.

**Note:** Only those methods usable with the number of control points you have defined will be displayed. If you don’t see a method, add more control points.

**Order**  The level of mathematical sophistication applied. Higher orders result in better matches but take more time and more control points. In numeric methods the highest order exponent used in the polynomial equations generated to transform the coordinate system of the target component. Enabled only for the **Numeric** method.

**Modify coordinate system**  Enabled when the **Simple** method is selected. If checked (default) the coordinate system (projection) of the component being georegistered is converted to the coordinate system of the guiding component. If not checked, the coordinate system is not changed.

**Scale pixels equally in X and Y direction**  When this option is off, the user can specify both the width and the height of the resulting image or surface component. When the option is turned on, the user can only specify the width of the resulting component and the system will automatically compute the height. By default the option is turned on.

**Interpolate pixels**  Enabled when georegistering non-palette images or surfaces using the **Numeric** method. Creates a much smoother image when transforming the image into the new projection. Very computationally expensive: requires approximately twice the processing time to georegister an image if enabled. **Note:** Methods other than **Numeric** always interpolate pixels for non-palette images and surfaces and never interpolate pixels for palette images.

**Save error surface using**  Create a surface containing the root mean square error value for each location in the georegistered component. Choose the data type for the error number saved in the surface at each location. If checked, this option doubles the time required for georegistration. Enabled for the **Numeric** method. See the Error Surfaces topic.

**Size**  The transformed size of the image or surface in pixels. By default this will be set to some value that attempts to approximately preserve the size of the image or surface before georegistration. Enabled for images or surfaces.

The **Order** chosen will have a great impact on the number of control points required, especially when using the **numeric** method. For numeric georegistration, there usually must be at least four times the order number in control points. Thus, for numeric registration of order 4, there must be at least 16 control points. This guideline is a minimum value. Certain arrangements of control points may require an even greater number of control points.

**Affine** registration works with fewer control points, as few as two or three in the case of **Affine (scale, shift, rotate).** However, greater care must be taken when assigning affine control points. In general, control points should be placed for affine registration so that control points are drawn along the outer border of an imaginary shape without control points in the middle of the shape. For example, placing control points in a rectangular or rhomboidal arrangement is OK. Placing control points in a circle with several additional control points in the center of the circle is not OK.

**Affine** registration is a good choice when registering scanned images of maps that have a graticule grid of latitude and longitude lines shown. One can choose four intersections of latitude and longitude graticule lines that are near the four corners of the image to use as control points. Because fewer control points are required affine registration is almost always a better choice if we can place control points in a rectangular or rhomboidal arrangement.
**Numeric** registration may be a better choice when registering images to control points that are scattered throughout the image. When using numeric registration, evenly distributed control points will yield better results. This will, however, usually require many more control points than affine registration.

Either affine or numeric registration is a good choice when registering scanned images of maps that have a graticule grid of latitude and longitude lines shown if we are able to place many control points. One can choose a number of intersections of latitude and longitude graticule lines to rapidly mark many control points.

In either case, if control points are restricted to only part of the image it is quite likely that other parts of the image will not be georegistered well. When used in maps they will appear to be out of alignment with overlying drawing objects. It is critically important to use control points that are dispersed throughout the entire image. With the numeric method, the more control points that are used, the better the ultimate georegistration.

Given the labor of marking many control points the appeal of **Simple** registration is obvious. If an image is already in a North up overhead view and need only be resized and moved to be registered to a given target drawing this is a good choice. If the image is in Orthographic (the default when importing from geographically-unaware formats like .jpg), re-project the drawing into Orthographic and note the central latitude and longitude. Use **Edit - Assign Projection** with the image to specify the same central latitude and longitude. Then use **Simple** registration to georegister the image to the drawing.

**Tech Tips**

Uncheck the *Autoscroll window on edit or selection operations* option in Tools - Options when working with control points, or use the *Require Alt key* so that no autoscroll happens unless the *Alt* key is pressed. Because using the *New Control Point* tool activates the mouse as soon as it leaves the Control Points pane, if this option is not deactivated the target window will begin to autoscroll as the mouse enters it.

Picking a control point within a component window sets the focus to that window.

Large images or surfaces can take a *very long time* to georegister when using **Affine (triangulation)** method or the **Numeric** method with higher **Order**. If desired, the georegistration process can be cancelled and then re-launched using a different method or lower **Order**.

Occasionally we might make a mistake in the placement or naming of control points that results in a bizarre and obviously wrong georegistration. In such cases, it is nice to know that **Undo** will work.

Images must be resized during some types of georegistration. Palette images will always be resized using the nearest neighbor method (no interpolation of colors) to guarantee that the georegistration process does not introduce any new colors.

**See Also**

- The File - Print dialog used to print a component includes a **Control Points** option that may be used to print the control points in a component.
- See the Georegistration topic for details on how to use control points to georegister images and drawings.
View - Panes - Errors

The Errors pane displays the description, module, line and line position of any errors that occur during the compilation phase of running a .NET script. Highlighting an error in the Errors pane and clicking Go To (or, simply double-clicking the error) will open the relevant script component and place the cursor at the location of the error.

The Errors pane will automatically pop open if any errors occur when running the Script - Compile to DLL command.

Go To - Open the relevant script component and place the cursor at the location of the error.

Show Module - Press in to show the module (script) names for each error.

Show Line - Press in to show the line number for that error within the script.

Show Line Position - Press in to show the position within the line for that error. Useful when there is more than one error on the same line.

Example

Suppose we start with the default VB.NET script seen above, which has been named VB.NET Hello.

We introduce an error by changing the first Shared to Snared.
When we compile the script using Compile to DLL an error will be reported and the errors pane will pop open (if it is not yet open) as seen above. Note that the highlighted error text is repeated at greater length in the status text readout at the bottom of the pane.

Double clicking on one of the errors, for example, the top error as seen above, will highlight the error line in the script.
Double clicking the middle error will jump to that line. As is typically the case with errors in programming, one error can generate more than one report of syntax errors as the problem cascades through the text to be parsed.

In this case, double clicking on the first error reported and the 'End Sub' error helps us rapidly identify the mistake.

Error Codes

Error codes reported in the Errors pane are generated by the Microsoft .NET compiler for that language. The easiest way of looking up an error code is to visit the Microsoft MSDN library at http://msdn.microsoft.com/library or http://msdn.microsoft.com and to search for the error code. Regrettably, not all Microsoft .NET compiler error codes are documented in the MSDN site.

For example, a search on BC30188 will fail to turn up an error code while a search on CS1519 (a C# error code often triggered by insertion of an extra curly bracket) will turn up an error code page.

See Also

Script - Compile to DLL
View - Panes - GPS Console

Manifold's GPS Console provides interactive control over a GPS (Global Positioning System) device connected to the computer via a serial port or USB connection. The GPS Console allows setting moving map mode or automatically fetching position and other data from the GPS device.

For details on using the GPS Console, see the Working with GPS Receivers topic.

Note: We must have at least one drawing in the project to accept data from the GPS device to use Track commands.

Controls

Connect / Disconnect - Press IN to connect to the GPS device using serial port parameters specified in the GPS properties dialog. The GPS dialog cannot fetch data from the GPS device until it is connected. Connecting to the GPS also serves as a test of the connection.

Track Center - "Moving map" mode: with each reading of data from the GPS receiver move the drawing so that the acquired position of the GPS is always in the center of the currently active window. When Track Center is engaged and either a Track or Track Sequence command is issued the active window will be panned to center on the location reported by the GPS.

Track Line - With each reading of data from the GPS receiver extend a line to the current position. This option creates a line object that shows the sequence of tracked positions.

Track Points - With each reading of data from the GPS receiver create a point at the current position. This option creates a series of points that show the sequence of tracked positions.

Track Columns - Used to choose which names will be used for destination fields of downloaded data. Otherwise, has no effect unless Track Points is selected. Raises a dialog allowing choice of the destination fields within the drawing's table for the various data fields acquired from the GPS receiver. Choose a blank destination in the Save To column to not import a given GPS field. When selected, Track Columns will write the selected data fields into the specified destination fields in the drawing's table with each reading of data from the GPS receiver.

Track - Press to fetch data from the GPS. Each press of this button commands one acquisition of data and the creation of objects or moving map action as specified by the Track Center/Line/Points/Fields buttons.

Track Sequence - Start / Stop acquiring data from the GPS receiver repeatedly using the given time interval between readings. When this button is pressed in, each specified time interval repeatedly commands data acquisition and the creation of objects or moving map action as specified by the Track Center/Line/Points/Fields buttons.

Download Waypoints - Enabled only when connected to a Garmin device using Garmin protocol or when an NMEA device is detected as sending waypoints. Press to download waypoints. Press again to cancel downloading. When pressed, the system will begin downloading waypoints from the GPS receiver, arranging them in order and creating a point object for each waypoint. A line object will be created for a route. To cancel the download process without creating any objects, click the Download.
/ Waypoints button again before the download is complete.

See the Waypoints and Projections paragraph below for important notes on projections.

Upload Waypoints - Enabled only when connected to a Garmin device using Garmin protocol. Press to upload waypoints. Press again to cancel uploading.

Properties - Set up the serial port connection to the GPS receiver by specifying either a USB connection or the serial port used and the serial protocol options required by the GPS receiver.

Drawing
Choose the drawing into which objects and data acquired from the GPS will be placed.

Refresh
Specify the time interval and time units (minutes, milliseconds or seconds) between automatic data acquisition when tracking.

Datum
Datum used by the GPS receiver. The datum setting dialog is resizable.

Altitude
Value reported by the GPS receiver for the height of the GPS antenna. Although handheld GPS receivers integrate the antenna into the instrument case, remember when using an external antenna the GPS device reports the position of the antenna.

Bearing
True heading computed by the GPS receiver for its motion based on differences between recent positions measured. The true heading is the magnetic heading corrected for magnetic variation reported for the current position by the GPS device.

Mag bearing
Magnetic bearing.

Mag variation
Magnetic variation.

Latitude
Current position latitude.

Longitude
Current position longitude.

Quality
Quality of the fix for the current position.

Speed
Speed in centimeters per hour computed by the GPS receiver for its motion based on differences between recent positions measured.

Satellites
The number of GPS satellites in view of the GPS receiver.

(Properties Dialog)

Device
Serial port or USB device to which the GPS receiver is connected. Most computers support COM1 and COM2 with connectors on the chassis. Manifold will automatically search the first 1000 possible COM port configurations and allow use of any COM port that appears to be working. If a USB port is used choose either the Garmin USB connection or the generic USB Human Interface connection if a non-Garmin USB GPS is in use. USB choices will not be available if a USB GPS device is not detected.

Baud rate
Speed of the serial connection. Use 4800 for most GPS receivers.
Data bits
Number of data bits in the serial protocol. Use 8 for most GPS receivers.

Stop bits
Number of stop bits in the serial protocol. Use 1 for most GPS receivers.

Parity
Parity setting for the serial protocol. Use None for most GPS receivers.

Test
Test connection to the GPS receiver. Instructs Manifold to listen for valid NMEA sentences.

Consult your GPS receiver’s user manual for information on what settings to use for the Properties dialog when using a serial port connection. The Properties dialog will fail to launch if there are no available COM ports or USB devices available. Unplugging a USB GPS device will automatically disconnect the GPS Console.

Pressing the Test button in the Properties dialog commands Manifold to try connecting to the GPS to see if it is alive and well and to report, if possible, what protocol the GPS recognizes. If a GPS is found, a message box will pop open to report whether it is using NMEA or GARMIN protocol. If it is using NMEA protocol, Manifold will report the NMEA tags recognized. If it is using GARMIN protocol Manifold will report a list of recognized GARMIN subprotocols recognized as well as the version of each subprotocol. If a GPS device switches from GARMIN to NMEA protocol, Manifold will automatically detect the change.

GPS Columns Dialog

The GPS Columns dialog allows specification of NMEA data that will be downloaded from the GPS, if available. Note that not all GPS devices will transmit all of the following fields. For example, only a depth sounder that repeats GPS location information while transmitting depth information as an NMEA repeater will provide GPS Depth information. The complete list of fields in the GPS Columns dialog is taken from all NMEA sentences that the Manifold GPS Console NMEA parser recognizes.

To capture a given field, double click into the Save To cell for that field and choose either the name of an existing column or to create a new column for that field choose [New Column].

Select All - Capture all fields.
Select None - Capture no fields.
Select Inverse - Uncapture all captured fields and capture all uncaptured fields. A fast way to use all but one field: click Select None, capture the one field not desired and then click Select Inverse.

GPS Air Temperature
Air temperature.

GPS Altitude
Altitude.

GPS Bearing
True bearing.

GPS Depth
Depth from the sounding transducer to the sea floor. Note that transducers are normally mounted some distance below the water line of a vessel, so that to compute the depth of water we may need to add the distance between the vessel's water line and the transducer to the reported GPS Depth figure. Some depth sounders reporting depth in an NMEA stream have a configuration option which allows adding a fixed distance between the water line and the transducer so that the GPS Depth reported will be the distance from the water line and not from the transducer.

GPS Dew Point
Dew point.
<table>
<thead>
<tr>
<th><strong>GPS Humidity</strong></th>
<th>Humidity.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GPS Humidity, Relative</strong></td>
<td>Relative humidity.</td>
</tr>
<tr>
<td><strong>GPS Latitude</strong></td>
<td>Latitude of current position.</td>
</tr>
<tr>
<td><strong>GPS Longitude</strong></td>
<td>Longitude of current position.</td>
</tr>
<tr>
<td><strong>GPS Magnetic Bearing</strong></td>
<td>Magnetic bearing (computed from true bearing using variation).</td>
</tr>
<tr>
<td><strong>GPS Magnetic Variation</strong></td>
<td>Variation used by the GPS for this position and date.</td>
</tr>
<tr>
<td><strong>GPS Quality</strong></td>
<td>Quality of the fix.</td>
</tr>
<tr>
<td><strong>GPS Receiving Frequency</strong></td>
<td>Receiving frequency.</td>
</tr>
<tr>
<td><strong>GPS Satellites</strong></td>
<td>Number of satellites used by the GPS to compute position.</td>
</tr>
<tr>
<td><strong>GPS Speed</strong></td>
<td>Speed of motion.</td>
</tr>
<tr>
<td><strong>GPS Stream Bearing</strong></td>
<td>Stream bearing.</td>
</tr>
<tr>
<td><strong>GPS Stream Magnetic Bearing</strong></td>
<td>Stream magnetic bearing.</td>
</tr>
<tr>
<td><strong>GPS Stream Speed</strong></td>
<td>Stream speed.</td>
</tr>
<tr>
<td><strong>GPS Time</strong></td>
<td>Time of measurement in Universal (Zulu) time.</td>
</tr>
<tr>
<td><strong>GPS Time, Local</strong></td>
<td>Local time</td>
</tr>
<tr>
<td><strong>GPS Time Zone, Local</strong></td>
<td>Local time zone.</td>
</tr>
<tr>
<td><strong>GPS Transducer to Keel</strong></td>
<td>Distance from the depth sounding transducer to the lowest point of the vessel. Transducers are normally mounted below the waterline but significantly above the keel. The distance from the transducer to the sea bottom must exceed this value to avoid grounding the vessel.</td>
</tr>
<tr>
<td><strong>GPS Transducer to Water Line</strong></td>
<td>Distance from the depth sounding transducer to the water line of the vessel. Transducers are normally mounted below the waterline but significantly above the keel. This distance should be added to the depth from the transducer to the sea bottom to get water depth.</td>
</tr>
<tr>
<td><strong>GPS Transmitting Frequency</strong></td>
<td>Transmitting frequency.</td>
</tr>
<tr>
<td><strong>GPS Water Temperature</strong></td>
<td>Water temperature.</td>
</tr>
<tr>
<td><strong>GPS Waypoint Name</strong></td>
<td>Used when downloading waypoint information.</td>
</tr>
<tr>
<td><strong>GPS Wind Bearing</strong></td>
<td>Absolute bearing of the wind.</td>
</tr>
<tr>
<td><strong>GPS Wind Bearing, Relative</strong></td>
<td>Apparent wind bearing considering the bearing of the vessel.</td>
</tr>
<tr>
<td><strong>GPS Wind Speed</strong></td>
<td>Absolute wind speed.</td>
</tr>
</tbody>
</table>

**See Also**

**Working with GPS Receivers**
View - Panes - History

The history pane displays a running history of key actions and results. The history pane persists throughout a session of Manifold, maintaining the history of events even if a new project is opened. Opening or closing a project or saving the active project will log the action into the history pane.

Controls

Freeze - Temporarily disables all output.

Save to File - Save recorded history to a text file.

Copy - Copy recorded history to the Clipboard.

Clear - Clear recorded history.

Print - Print recorded history.

Clearing the contents of the history pane will automatically disable Copy, Print, Save to File and other toolbar buttons working with history text.

Manifold can log information to the history pane such as the time required to execute functions. We can control what information is logged using the Logging options in the Tools - Options dialog.
View - Panes - Info
Reports data attributes of objects in drawings and relevant information for some other components.

Drawings

When a drawing window or a drawing layer in a map has the focus, the info pane shows data from either all objects or only those objects in the selection, showing information for one object at a time. The scope of the info pane is either the set of all objects or the set of selected objects. The context object is the one shown in the info pane.

By default the info pane shows all objects with the context object being the first object. We can step through each object using the First, Next, Previous and Last buttons or by clicking on an object in the drawing to select it.

Center - Center the context object in the window.

Zoom - Center the context object in the window and zoom to fit the object. Zooms are most effective with area or line objects. To distinguish between points, either use Center or Zoom and then zoom in and Ping to see the exact object.

Ping - Highlight the context object by flashing areas and lines and by drawing converging circles about points. Pinging very small lines or areas will also show the location by drawing a series of converging circles. A great way to see what the context object is.

First - Show the first object in the scope as the context object.

Previous - Show the previous object in the scope as the context object.

Next - Show the next object in the scope as the context object.

Last - Show the last object in the scope as the context object.

Filter Selected - Only show selected objects.
**Example**

We import the example drawing of Mexican provinces from the **Examples** folder on the Manifold CD.

We open the drawing and open the info pane using **View - Panes - Info**. We press in the **Filter Selection** button on the info pane toolbar.

Initially, no objects are selected so the info pane shows nothing because there is nothing in the selection.

Using the Query toolbar we select all provinces with a population less than **800000**.

Immediately, the provinces with populations less than **800000** appear in the drawing in red selection color.
The first object in the selection, the province of Baja California Sur, appears in the info pane. If we would like to step through the rest of the selected objects we can press the **Next** button in the toolbar.

Each press of the **Next** button will bring the next object in the selection into the info pane.

To see which of the selected objects is the province of Tlaxcala we can press the **Ping** button to call attention to the Tlaxcala province by flashing the Tlaxcala area object.

Very small areas may be difficult to locate even when they are flashing. We can always press the **Zoom** button to zoom the Tlaxcala province to fit the display window.
This makes it clear where the Tlaxcala object is located. We can zoom out in steps to see the region surrounding it, if desired.

**Profiles and Elevations**

Profiles and elevations are optional components that are available when the Surface Tools package is installed. See the Profiles and Elevations topic for more information.

When the focus is on a profile or elevation window or on a profile layer in a map, the info pane will report the average height for the profile as well as the minimum and maximum height in the profile. The number of decimal digits used after the decimal point is set by opening the elevation or profile and using the View - Display Options dialog to change the number of digits desired for that elevation or profile.
View - Panes - Layers

The **Layers pane** is used with mostly with map windows to control the stack of drawings, images and text components that appear in a map. It is also used with other components, for example, when an image window or drawing window or a layout window has the focus:

- With an image window, the layers pane shows the border, background and image channels so that any of these items may be turned on or off.
- With a drawing window, the layers pane shows the background and also any views of that drawing used in layouts.
- With a print layout window, the layers pane shows individual elements in the layout to enable them to be turned on and off.

**Using the Layers Pane with a Map Window**

The layers pane displays the layer stack in a map. The layers pane is more convenient to use with many layers than the tab controls at the bottom of the map. For intensive image editing work involving many layers it provides faster control over layer characteristics (such as transparency) as well.

**Layers Pane Controls**

- Click on a layer to highlight it and to make it the active layer.
- Use the checkboxes to turn layers on and off for display.
- Each image layer may be set to a percent opacity to specify partial layer opacity.
- Move layers up and down in the layer stack with the **Move Up** and **Move Down** buttons.
- The **Border** and **Background** may be turned off and on using their checkboxes.

- **Switch To** - Make the highlighted layer the active layer in the map. Double clicking on a layer also will make it the active layer.
- **Show All** - Show all layers in the map.
- **Hide All** - Do not show any layers in the map.
- **Move to Top** - Move the active layer to the top of the layer stack.
- **Move Up** - Move the active layer up one position in the layer stack.
- **Move Down** - Move the active layer down one position in the layer stack.
- **Move to Bottom** - Move the active layer to the bottom of the layer stack.
- **Delete Layer** - Remove this layer from the map.
- **Restrictions** - Show and edit layer restrictions for the highlighted layer.
- **Opacity** - Show opacity controls for layers.
- **Properties** - Show properties for this component. Equivalent to opening the component in its own window and choosing View - Properties .

**The Border and Background Layers**
The layers pane will always show two additional layers: the **Border** layer and the **Background** layer. These are system layers and are not real "layers" in that they do not correspond to any drawing or image in the project. Both may be switched ON or OFF via their checkboxes.

The **Background** shows what color to put underneath all other layers. By default white, it may be switched off to easily see if any white space in the map is part of the background or not.

The **Border** shows the proposed size of the image for printing to a printer or to a file. It may also be used as a guide when merging images into each other. The **Border** can be shown in various styles, either as a simple border, as a border with all external areas opaque (to get a better idea of how an image cropped down to the frame size will look), or as a partially transparent matte in regions outside the frame. This latter setting gives an idea of what the final cropped image will look like while still allowing a view of items outside the frame to use as a guide when moving layers about to bring items into view.

### Setting Background Color

Background color for all components is set by default in the Tools - Options dialog. Each individual component can have its background color set by opening the component in a window and choosing View - Properties.

### Layer Restrictions

The **Restrictions** button in the layers pane displays and sets layer restrictions. Layer restrictions may also be set by right clicking on a layer tab in a map window and choosing Restrictions.

Each layer within a map may be specified to be clickable, editable or selectable (in any combination) with mouse commands. Each layer has three restriction options:

- Enable mouse clicks
- Enable mouse editing
- Enable mouse selection

By default, all three options are enabled so objects in layers by default may be clicked, edited or selected.

The layers pane shows edit restrictions with a no marking each restriction. The "yes" implied by default is not shown for greater legibility. For example, in the above illustration the **Cities** layer is clickable but not editable or selectable. Double-click into any restriction setting in the layers pane to toggle it between no and the default yes setting.
If we were to right click on the Cities layer tab in the map and choose Restrictions the Layer Restrictions dialog would show restrictions status for that layer. In this dialog a check mark enables the function and no check mark removes the function. For example unchecking the Enable mouse edits box will prevent a CTRL-ALT click on an object in that layer from selecting it for editing.

Layer restrictions are used to show layers in maps while preventing some layers from participating in mouse commands. This makes it easier to edit desired layers in complex maps. Another use for layer restrictions is in Manifold IMS to restrict the action of hyperlinks or the Info tool to a limited set of layers.

Layer restrictions operate only with mouse commands. Keyboard and menu commands are not limited by layer restrictions. For example, if a layer has had the Enable mouse selection checkbox unchecked none of the mouse selection commands will select objects in that layer. However, using CTRL-A or Edit - Select All will select all objects in that layer, the query toolbar will select objects in that layer and queries will select objects in that layer.

Layer Opacity

Image layers may be given an opacity percentage that applies to the entire layer. Zero percent is no opacity (totally transparent) while one hundred percent opacity makes the layer totally opaque. If an image layer contains an RGBA image where individual pixels may have different degrees of transparency, the layer opacity will be evenly combined in a proportional way with all pixels.

To set layer opacity, press in the Opacity button and an Opacity column will appear in the layers pane. Double-click into the opacity value for each layer to change it. Enter a value from 0 to 100 (no need to enter the % character) and press Enter.

Using the Layers Pane with Drawings

The Layers pane is used to control the appearance of drawings within drawing windows. The layers pane includes checkboxes for the background and border. The border "layer" with drawings shows an enclosing box about the widest extents of objects in the drawing.

By default, drawings are shown using the checkerboard background Manifold uses to provide a backdrop for any transparent regions. The layers pane is shown to the right of the drawing window.
Checking the **Background** box in the layers pane will replace the checkerboard background with whatever is the default background color.

Checking the **Borders** box will draw a one-pixel border that represents the minimum enclosing box for all objects in the drawing. When working with drawings that include points, using the **Border** is a handy way to see if any very small objects exist far away and unnoticed from other objects, since the border will expand to include all objects.

The Layers pane will also show views of the drawing used in print layouts. See the Specifying Views in Layouts topic for details.

**Using the Layers Pane with Images**

The **Layers pane** shows all of the channels in the active image. The appearance of pixels comes from blending primary colors called **channels**. Channels and their use via the Layers pane are described in detail in the Images and Channels topic.

![Layers pane](image)

RGB images consist of three channels: red, green and blue.

By checking or unchecking the channel boxes we can turn each channels ON or OFF for display. For example, checking only the channel for blue and unchecking red and green channels will display a standard RGB image using only the blue tones of which it consists.
RGBa images have an additional channel called the **Alpha** channel that specifies transparency for each pixel.

If we have an RGBa image like the one above,

We can uncheck the Alpha channel.

Doing so removes the influence of the alpha transparency channel and so displays the image as an ordinary RGB image without per-pixel alpha transparency applied.

Compressed images will show R, G and B layers in the layers pane. If channel data from a multi-spectral image has been assigned to the alpha channel for a compressed image using the View - Display Options pane, then an alpha channel layer will appear in the layers pane as well for a compressed image.
See the Images topic and the Images and Channels topic for additional examples using the checkboxes.

**Grayscale and Palette Images**

Both grayscale and palette images have one channel:

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Border</td>
<td>Palette</td>
<td>Background</td>
</tr>
</tbody>
</table>

A grayscale image is simply a palette image with a grayscale palette. For both grayscale and palette images the single palette channel will be shown in the Layers pane for these image types.

**Border and Background**

The border and background "layers" appear in the layers pane for images. These checkboxes control the appearance of a border frame around the image as well as the appearance of background color. The screen shots below illustrate the border and background layers pane elements using our sample bronze image. In the examples all pixels other than those in the monument itself have been selected and then deleted so that they are invisible pixels.

The default appearance of invisible pixels is complete transparency, which shows the checkerboard backdrop of all Manifold windows. Without a border it is difficult to see where the image ends.
Checking the **Border** box will turn on a one-pixel wide black border about the edges of the image. This immediately lets us see the horizontal and vertical size of the image.

Checking the **Background** box fills in the image window with background color. This shows us what the image will look like if printed or rendered into a graphics format like .gif or .jpg that does not support alpha transparency.

On a white background it may be difficult to tell where the actual image begins and ends. We can check both **Border** and **Background** to see how the image looks as well as seeing the extent of the image.

**The Layers Pane and Layouts**

When a print layout window has the focus the layers pane will show each item in the layout as a layer. Available commands include:
- Checking a box will turn the layout element on and unchecking the box will turn the layout element off. Hiding an element by unchecking its box will prevent it from being clicked, selected or printed.
- Moving "layers" up and down in the layers pane will move the layout elements up and down relative to each other within the layout.
- Double clicking onto a text label in the layers pane will allow immediate editing of the text in the box directly from the layers pane.
- The **Delete Layer** button in the layers pane toolbar will be enabled when any layer except the main layer is clicked. Deleting a layer will remove this element from the layout.
- The **Properties** button in the layers pane toolbar will open the properties dialog for any component-based element in the layout.

The layers pane is a key control when working with complex layouts.

In the example above we see a layout together with the layers pane. The layout is based on a main element that is a drawing of Europe. Below it is a text element that consist of the phrase "Copyright 2002" repeated several times in an outline font using a light gray color. Above the main element is another text element giving the title, "Europe," and two images, the example **Bronze** and **Schloss** images. The **Background** layer is turned off, so behind the layout page we see the checkerboard pattern used in Manifold when there is no background.

Note that the layout elements added with layout tools, such as text, are shown in the layers pane with special "layout element" icons. Text elements will be named using the initial characters of their text.

With many overlapping elements in a layout we can use the layers pane to turn off upper elements if we want to select or otherwise work with lower elements.

**Layouts in the Layers Pane with other Components**

If a component has any Layouts created they will appear as "layers" in the layers pane for that component. Checking the box for one of these print layout layers will cause a layout rectangle to appear in the component that shows the region covered by the layout.
In the illustration above the drawing has four layouts that show different parts of Mexico. Three of the layouts have been checked in the layers pane causing three layout preview rectangles to appear in the drawing.

The preview layout rectangles will be automatically adjusted if we change the scope of the layout.

For example, suppose we open a drawing window and a layout window for the same drawing and show the layout in the drawing window by checking the layout layer’s checkbox. We select an area in the drawing by clicking on it with Select Touch. The scope of the layout (set by right clicking the layout in the layout window and choosing Properties) is set to entire component.
If we now change the scope of the layout to be the selection, the layout will show only the selected objects (one area, Germany, in this illustration) and simultaneously the drawing window will be updated to show the new region covered by this area.

Right clicking onto the hatched border of one of the layout rectangles in the drawing will cause a context menu to appear with controls based on that layout rectangle. For example, we can Zoom to a given layout rectangle, Print it or change its Properties. If a layout is empty (for example, if the layout scope is set to selection and nothing is selected in the parent component) zooming to the layout will do nothing.

Use Tools - Options - Colors - Layout Rectangle to change the color in which layout rectangles are shown. The default color is black.

Notes

One way of "seeing" invisible pixels is to use the Selections pane built-in invisible pixels selection and to push in the Preview button in the Selections pane to see any invisible pixels in blue preview color. However, invisible pixels are not the only way to render pixels invisible in Manifold images. If the image is an RGBA image it is possible to select some pixels and to set their alpha channel transparency very high so that they are completely transparent. In this case they are not invisible pixels and will not appear in the invisible pixels system selection in the Selections pane.

Only maps can have true "layers" in Manifold in the sense that they can layer more than one component within the same map window. Neither the border and background "layers" in the Layers pane for drawings and images, nor the channel "layers" in the Layers pane for images are true layers even though they appear in the Layers pane in the same manner as do layers in maps. These are simply system controls that take advantage of the Layers pane as a conceptually convenient user interface.

Layers pane settings apply only to the context window, that is the window that has the focus. For example, suppose we have the same RGB image component open in three image windows and also in view in two opened map windows. If we click on one of the image windows to give the focus and turn off the Red channel in the Layers pane, the image will immediately change appearance in the context window but will not change appearance in any of the other opened windows. If we then click on one of the other image windows the Layers pane for that window will still have the Red channel turned on.

See Also:

Images and Channels
RGBA Pixel Transparency
Masks
Painting within Channels
Separating Images by Channels
Combining Channels into Images
Layouts
View - Panes - Notes

The Notes pane provides a place for making text notes about components and to capture component-specific information. As the focus changes from one component window to the other, the contents of the Notes pane will change to the notes made for that component.

- **Load From File** - Load the contents of a text file into the Notes pane.
- **Save To File** - Save the contents of the Notes pane into a text file.
- **Clear** - Delete all text from the Notes pane.
- **Print** - Print the contents of the Notes pane.

The Notes pane is often left open to a height of a few lines when working with many different component windows. It's then easy to click into the Notes pane and to jot down a few words describing work done, "to do" items for that component and so on. When the project is saved all notes made will be saved as well.

After pasting a drawing or table as a surface when the surface is opened the Notes pane will list the interpolation parameters that were used to create the surface.

The Notes pane will contain EXIF tag information imported for jpg images if the images contain any EXIF tags.
View - Panes - Project

The project pane shows all components in our project. It also shows how components are associated, such as a drawing and its table. We can use the project pane to open components in windows, to drag and drop components into maps, to rename components and to organize components within folders. The status area at the bottom of the pane will provide summary information about any highlighted component.

We can also use the project pane to copy and paste components. Copy and paste within the project pane is used not only to make copies of components, but also to transform one type of component into another. For example, we might copy a geocoded table and paste it as a drawing, or we might copy an image and paste it as a terrain. When using Copy and Paste As to transform one type of component into another, supplemental dialogs will be raised as necessary after the Paste As command.

Click on a component to highlight it. The project pane’s toolbar commands will then apply to that component.

- **Open** - Open the highlighted component in a window. If it is already open, activate the opened window and move it above other open windows.
- **Open in New Window** - Open the highlighted component in a window. If it is already open in a window, open it in a new window.
- **Run** - Run the highlighted component. This command is normally used to run an executable component, such as a script.
- **Cut** - Copy the highlighted component to the Windows clipboard and then delete it from the project.
- **Copy** - Copy the highlighted component to the Windows clipboard.
- **Paste** - Paste the Windows clipboard into the project.
- **Paste As** - Paste the Windows clipboard into the project as a given component. The down arrow pulls down a menu showing what types of components can be used for the data on the clipboard.
- **Delete** - Delete the highlighted component.
Create - Create new components, such as folder, drawings, images, etc. in the project.

Refresh - Appears in Enterprise Edition. Press to refresh the project display immediately in advance of any autorefresh time period, polling all Enterprise servers to see if any shared component has been checked out or checked back in.

CTRL- * - Pressing CTRL- * (Asterisk) with the focus on the Project pane expands all components.

CTRL- / - Pressing CTRL- / (slash) with the focus on the Project pane collapses all components.

Context Menus in the Project Pane

Right clicking on a component will call up a menu of useful commands. For example, the context menu for most components will include an Export command. See the Project Pane - Context Menus topic.

Folders

The create folder button in the project pane toolbar creates folders within the project. Use folders to organize components within the project and to keep the project pane manageable.

This is especially important when importing drawings from complex file formats such as those used with VMAP and TIGER/Line, where a single import will create numerous drawings and maps. Imports will create components within the current folder. Before importing from such formats, create a new folder in the project to highlight it and then choose File - Import. Imported components will be created within the folder specified.

To move files between folders, drag components and drop them into a folder (drop them onto the folder icon itself and not the hierarchy beneath the folder). Folders may be created within other folders. If a folder is highlighted in the project pane and the create folder command is issued, the new folder will be created within the highlighted folder. Folders may be dragged and dropped into other folders. To move a folder out from within another folder, drag it and drop it into the "white space" of the project outside another folder.

See Also

Copy and Paste As
View - Panes - Review

The Review pane shows conflicts arising from concurrent multi-user editing of linked drawings, displaying the Manifold object ID of each conflicted object and the conflict, with area, line or point in the Type column indicating an editing conflict. The Review pane works when editing drawings or drawing layers in maps.

To use the Review pane, we must have Enterprise Edition and have also specified a version column in the database table to which the drawing has been linked. In addition, the Edit Mode of the drawing in the Link / Share Status dialog must have been set to Review changes made by others (the default when a version column has been specified). See the Multi-User Editing of Linked Drawings topic for details.

Click on a conflict to highlight it. The Review pane’s toolbar commands will then apply to that component. The drawing window will show the local version of each object conflict in blue and the remote version in red (colors may be set in the Tools - Options dialog). The easiest way to remember the colors is to think that we are always right (of course!) so that changes made by other users are shown in an alarming, red color.

The local version is the object as we have edited it in our project. The remote version is the object as someone else has edited it.

Controls

- **Center** - A “go to” command to center that conflict in the drawing window.
- **Zoom** - Zoom the drawing so that the conflict is completely visible.
- **Use Local** - Commit local changes and update the database to use the local version of the object for the highlighted conflict. Discard changes made by others.
Use All Local - Commit local changes and update the database to use the local version of the object for all conflicts. Discard changes made by others.

Use Remote - Discard local changes and update the database to use the remote version of the object for the highlighted conflict. Download remote changes into the local drawing.

Use All Remote - Discard local changes and update the database to use the remote version of the object for all conflicts. Download remote changes into the local drawing.

View Conflict - Enable (that is, press in, the default setting) to show conflicts in the drawing window using red color and blue color for object geometry and server formatting or local formatting for format conflicts. When not enabled, object conflicts will not be previewed in the drawing window.

(Conflicts pane) A list of editing conflicts and their types detected from editing changes made by other users which conflict with editing changes we make.

Conflict - Either the object ID for editing conflicts involving the metric of objects or the name of the formatting characteristic for a formatting conflict.

Type - Either Area, Line or Point for editing conflicts to indicate the type of object involved or Format to indicate a formatting conflict.

(status bar) A text line at the bottom of the dialog indicating the number of editing conflicts.

Workflow

When presented with a list of conflicts we normally highlight each in turn and then use Center and Zoom to bring each conflict into view in the drawing window. Based on what we see we then decide to use either the local version or the remote version of the object.

If we are feeling particularly authoritative, we can force usage of our edits for all conflicts by choosing the Use All Local command. Conversely, if we think everyone else's edits are to be preferred to ours we can choose the Use All Remote command. Either of these commands should be used with care.

Projects may contain many linked drawings, which could be linked from different data sources. The Review pane will show conflicts for whatever drawing window or drawing layer has the focus.

Reviewing Formatting Conflicts

If the Administrator Console has been used to enable formatting storage on the database server for a linked drawing then linked drawings will show formatting when they are linked into a project and any changes in formatting will be saved back to the database.

Different users can edit the same linked drawing at the same time. If formatting is enabled, different users can change formatting at the same time. Whenever a user changes formatting the new formatting is immediately sent back to the database. Other users working with that same linked drawing will see those formatting changes when they refresh data.

It's a good idea, therefore, to do a Refresh Data command for a linked drawing before changing formatting, to see if someone else has already made any formatting changes not yet displayed. However, even if we do a Refresh Data command it is possible that someone else changes formatting while we are in the process of making formatting changes ourselves and that the other person accomplishes their changes ahead of ours. In that case, an editing conflict will occur. The Review pane will display editing conflicts over formatting and can be used to resolve them.
Selecting an editing conflict with formatting data in the Review pane previews the server version of the formatting in drawing, map and theme windows (the server version of the formatting is applied on top of whatever formatting has been specified locally, to make the difference between formatting as saved on the server and local formatting more noticeable).

**Example**

In this example we consider formatting changes made to the same linked drawing on two different systems. The drawing used is a version of the **US Main** sample drawing that has been exported into an Oracle database. The drawing has had formatting storage enabled through the use of **Administrator Console**. The systems are known as the **Blue** system and the **Green** system, from the color of the different Windows XP color schemes used on the different systems.

On the **Blue** system we open the linked drawing as seen above. The drawing uses a thematic format based upon a **Color** field to color states in the US. California has a pink color because pink color has been assigned in the thematic format for use with the value in the **Color** field associated with California.

On the **Green** system we also open the linked drawing as seen above. Because formatting has been enabled in the Oracle database, whenever we link this drawing into a project it will show the formatting stored with the drawing.
Let's go back to the **Blue** system and change the thematic format.

We will click into the area background color well (the thematic formatting is used for area background color) to launch the thematic formatting dialog. We then click into the pink color well to change the color.

We will change it to a pastel shade of blue-green.
When we press OK in the thematic formatting dialog the drawing will show the new formatting and the change in the thematic format will be automatically uploaded into the database.

It is important for beginners not to get confused over the subtle, but very significant, difference between a thematic format, which specifies formatting, and values of data attributes in a drawing’s table that are used by that thematic format. Changing the color used by a thematic format as seen above is a change in formatting. Changing a data attribute in a drawing’s table is just changing a data attribute, not changing formatting, even though as a result of following the formatting in force it may change the color of a particular object being displayed.

Suppose, for example, that we decided to change the color of the state of California by double-clicking into the drawing onto the California area object to change the value of the Color field for that area. We can see from the screen shots above that pink color had been assigned to a Color value of 1 and gray color had been assigned to a Color value of 3. Had we double-clicked into California and change the Color value from 1 to 3 that would have changed the color used to display the state to gray.

However, that would not have changed the formatting because the thematic format defined for that drawing would still be the same: objects with a Color field of 1 would continue to be drawn in pink and objects with a Color field of 3 would continue to be drawn in gray. It is just that the mix of values for Color for different objects would have changed.

When data attributes in tables are changed for linked drawings those changes are immediately communicated into the database and are not subject to review or resolution as are editing changes in object geometry or in formatting. This is conceptually a difference from the way editing changes in geometry or formatting are handled, but it is the long-standing tradition in multiuser database work to do it that way.

Let's get back to the main theme of this example, the change in formatting done on the Blue system and what effect it has, if any, on the Green system.
Back on the **Green** system, as seen above, we can see that the formatting changes made in the **Blue** system do not appear. They will not appear until we do a **Refresh Data** to synchronize the local display with whatever is already on the server. Suppose we do not do a refresh, but instead go ahead and make formatting changes on the **Green** system?

If we open thematic formatting for area background color on the **Green** system we see that the color well for a **Color** value of 1 is still pink. We click into it to change the color.

Let's change the color to a pastel shade of purple and then click **OK**.
As expected, the color of all areas with a Color value of 1 changes to purple. The Review pane will show a formatting conflict.

We will rearrange our screenshot to show both the Review pane and the drawing window as seen above. Note that the Review pane shows a Format conflict for Area Background. By default, the Review pane shows all conflicts. Until a particular conflict is highlighted, the local display will show local formatting. When we click onto a conflict to highlight the display will show what the version on the server looks like.

When we highlight the conflict the display changes to show the formatting stored on the server, which was set by the user on the Blue system. We can switch back and forth between display of server formatting and local formatting by pressing the View Conflict button to enable or disable it.
With the **View Conflict** button popped out the display will show local formatting. To make a choice between use of server formatting or local formatting we can press one of the **Use** buttons.

For example, pressing the **Use Local** button tells Manifold to resolve the formatting conflict by choosing the local formatting and uploading that into the server to override what was placed there by the user on the **Blue** system.
This resolves the editing conflict by making the server version the same as the local version. The Review pane now no longer shows any editing conflicts.

Note that if we were to now go to the Blue system we would still see the (now obsolete) formatting using pastel blue-green color until we do a Refresh Data command to update the display.

See Also

Multi-User Editing of Linked Drawings
View - Properties - Link / Share Status
View - Panes - Selections

The View - Panes - Selections pane is used to save selections for later use. If we select a particular region of pixels in an image, for example, we can save that region so that later, if desired, we can select exactly the same pixels once more. The Selections pane is enabled when the focus is on any component (image, drawing, chart or table) in which selections can be made.

Each component has its own set of saved selections. We can save up to seven selections for each drawing and up to six selections for each image (one saved selection in images is reserved for showing any regions of invisible pixels leaving only six available slots for selections saved by the user). Maps can have as many saved selections as exist within the various drawing and image layers that comprise the map.

When the Selections pane is open, switching focus to a different window will switch the contents of the Selections pane to the saved selections for the component(s) shown in that window. The selections pane will not be enabled if a window from which selections can be made and saved does not have the focus (is the active window or layer).

To Save a Selection

1. Make a selection.
2. Open the Selections pane and press the New Selection button. The selection will be added to the saved selections list as "Selection1"
3. Double click onto the saved selection name to rename it, if desired.

To Use a Saved Selection

1. Click on the target component to make it active. Open the Selections pane.
2. Click on the desired saved selection to highlight it. Checking the Preview box will show that saved selection in blue color in the component so you can see how it relates to any selection currently made.
3. Click on one of the selection command buttons to combine the saved selection with any selection that is currently made. For example, clicking on the Subtract from Selection button will take the highlighted saved selection and will subtract the pixels or objects it contains from the current selection.

To Delete a Saved Selection

1. Click on the desired saved selection to highlight it.
2. Press the Delete Selection button.

Selection Commands

Five selection command buttons are arrayed at the top of the Selections pane. These commands apply to whatever saved selection has been highlighted in the Selections pane. If we click on a saved selection to highlight it and then press one of these buttons, it will be combined with whatever is the current selection in the image. The selection commands allow us to use the saved selection to replace the selection, to add to the selection, to subtract from the selection and to intersect the selection.

Replace Selection - Replace the existing selection with the highlighted saved selection.

Add to Selection - Add the highlighted saved selection to the existing selection.

Subtract from Selection - Subtract the highlighted saved selection from the existing selection.

Invert with Selection - Deselect what is in the highlighted saved selection that was already in the existing selection and otherwise add the highlighted saved selection.
**Intersect Selection** - Select only the region of overlap between the highlighted saved selection and what is in the existing selection. If there is no overlap, nothing will be selected.

**New Selection** - Save the current selection as a new entry in the Selections pane list.

**Delete Selection** - Delete the highlighted saved selection from the Selections pane list.

**Preview** - Press to see a preview (in blue color) in the drawing or image window of the highlighted saved selection.

See the Selection topic for examples.

**Invisible Pixels**

Images in Manifold can have invisible pixels that do not appear in the image. They are simply "placeholders" for empty regions of the image. When images containing invisible pixels are stacked above one another in a map, items in layers below the invisible pixels will be visible.

Invisible pixels are very frequently used when composing advanced images (that is, maps) that consist of stacks of many constituent image components. When working with many different image components that contain transparent regions, especially when combined with images that have white pixels or drawings with "white space" in them, it is easy to forget which parts are transparent and which regions are simply "white space" or white pixels seen through the layer stack.

In addition, we often would like to use the region of invisible pixels in selection combinations with selections made in other parts of the image. To support this usage and to make it easy to always rapidly see the region of invisible pixels, Manifold devotes one of the seven saved selections in images to a saved selection that always shows the region of invisible pixels. This saved selection will appear in the Selections pane if any invisible pixels occur in the image. It cannot be removed from the Selections pane nor can it be renamed.

To see the region of transparency, simply click on this saved selection to highlight it and check the Preview box. It will appear in blue preview color.

See the Invisible Pixels and Selection topic for information on selecting invisible pixels and on using the Invisible Pixels selection in the Selections pane.

**Saved Selections and the Transform Toolbar**

Saved selections will appear in transform toolbar boxes that can work with selection sets. For example, if a drawing has two saved selections called *South West States* and *Mountain West* then these will appear as choices in the transform toolbar when that drawing has the focus.

When drawings appear together in a map, if any drawing layer has the focus the transform toolbar for the map will list all saved selections in all drawings. This is a great convenience, but it also leads to a slight complication in that all saved selections using the same name are treated alike. Saved selections in the same drawing are required to have different names, but saved selections in different drawings might use the same name.

The transform toolbar for the map will treat all drawing saved selections using the same name as one, combined saved selection. For example, if we have two drawing layers in a map, one called *East* and the other called *West* and they both have a saved selection called *Cities*, then using the transform toolbar to create a convex hull using the saved selection choice called *Cities* will use the objects from both the *East* and *West* drawings that appear in their respective *Cities* saved selection.

Therefore, it is a good idea to use unique names for saved selections in drawings that might appear together in the same map. For example, we could name the saved selection in one drawing *East - Cities* and that in the other drawing *West - Cities*. 
See Also

Selection
View - Panes - Tool Properties

Displays settings for the current tool in use, for example, threshold for Select Touch when used with images. The Tool Properties pane is an essential control when editing images with painting tools found on the Tools toolbar.

Tool Properties and Images

When an image window has the focus the Tool Properties pane sets options used with painting and selection tools for images. See, for example, Editing Images.

- **Snap tolerance**: When a snap mode is on, the distance in pixels or physical units within which the cursor must be to a given grid or graticule snap location when snapping to grid or graticule.

- **Value tolerance**: Used with selection, paint bucket fills, etc. Specifies how close a color must be to the color specified before it is considered the same color. Increasing the tolerance value will allow colors that are less and less similar to the touched color to be considered the same.

- **Density**: A technical parameter that defines the relative height of the density curve used for tools that paint pixels using a curve to define the amount of paint applied. Similar in visual effect to the "hardness" tool parameter in PhotoShop.

- **Opacity**: The opacity with which a "Paint" tool is applied. Lower opacity means the tool will not fully cover the pixels it overwrites. Does not apply to tools other than the "Paint" tools, for example, it does not apply to the Brush, Bucket or Gradient tools.

Tool Properties and Tables

When a table window has the focus the Tool Properties pane sets options for use with table operations such as Transform Toolbar operators. Token operations in particular will often use the Separate tokens with setting. Separators are defined in this pane because they are frequently changed during token operations so it is more convenient to have them available in pane that remains open than within an options dialog that is not always visible.

- **Token separators**: Characters to be used to separate tokens. By default the space character, tab, carriage return and newline characters are used as token separators.

- **Week start**: Day that is reckoned to be the beginning of the week. Used in date calculations by Transform toolbar operators such as Copy Day of Week from and similar.

- **Angles**: Units of measure for angular measurements.

- **Number of days in the first week of a year**: The number of days of the year that a week must contain before the year is reckoned to have started with that week. Set to 1 by default, but can be adjusted for different "rounding" of years in calendar calculations. Used in date calculations by the Copy Week from operator in the Transform toolbar.

- **Ignore case**: Treat upper and lower case letters the same in text fields.

- **Ignore leading and trailing whitespace**: Ignore any whitespace characters occurring before or after other characters in text fields. If checked, "Paris" would be considered identical to "Paris".

- **Ignore interior**: Ignore any whitespace distributed within a sequence of
whitespace non-whitespace characters. If checked, "San Luis Obispo" would be considered identical to "SanLuisObispo"

Tool Properties and Surfaces

When a surface window has the focus the Tool Properties pane sets options used with surfaces, namely snap tolerance.

- **Snap tolerance** When a snap mode is on, the distance in pixels or physical units within which the cursor must be to a given grid or graticule snap location when snapping to grid or graticules.

- **Value tolerance** Color tolerance to use to reckon that pixels are the same color as the pixel originally clicked.

Tool Properties and Drawings or Themes

When a drawing or theme window has the focus the Tool Properties pane sets options used with drawings, namely snap tolerance.

- **Snap tolerance** When a snap mode is on, the distance in pixels or physical units within which the cursor must be to a given snap item (such as a line when snapping to lines) before the cursor snaps to that item.

Tool Properties and Drawing Layers in Maps

When a drawing layer in a map is active the tool properties pane is configured to work with the tracing tools. See the Tracing Tools topic.

- **Snap tolerance** When a snap mode is on, the distance in pixels or physical units within which the cursor must be to a given snap item (such as a line when snapping to lines) before the cursor snaps to that item.

- **Value tolerance** Color tolerance to use to reckon that pixels are the same color as the pixel originally clicked.

- **Point size** A distance parameter given in native drawing measurement units that is used to distinguish a point from an area. Pixel regions smaller than this distance will be created as points, while those larger than this distance will be created as areas.

- **Match neighbors** When automatically creating areas, fill in the area to align exactly to the edge of any neighboring areas that already exist.

- **Trim size** A distance parameter given in native drawing measurement units that is used to trim unwanted side lines when automatically creating areas.
View - Panes - Variables

The Variables pane is enabled when the Manifold Debugger is installed.

The Variables pane shows the value of variables within scripts as they are executed together with information about the context in which those variables appear. It may also be used to change the value of variables for debugging purposes. Variables are shown within a hierarchical tree diagram.

- **Show Type** - Press in to show the type of the variable.
- **Show Common Properties** - Press in to show the Application and Parent properties that are common to all Manifold objects. Also controls the display of the Item and _NewEnum properties normally found in collections.

For example, suppose we’ve stopped execution in a script called A at a breakpoint where another script is run.

```java
Sub Main
    Set components = Application.ActiveDocument.ComponentSet
    Set anotherScript = components(components.ItemByName(“B”)).anotherScript.Run
End Sub
```

With both Show buttons pressed in, the Variables pane will report the following values:

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>components</td>
<td>Object</td>
<td></td>
</tr>
<tr>
<td>anotherScript</td>
<td>Object</td>
<td></td>
</tr>
<tr>
<td>Application</td>
<td>Object</td>
<td></td>
</tr>
<tr>
<td>Parent</td>
<td>Object</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>“B”</td>
<td>String</td>
</tr>
<tr>
<td>Description</td>
<td>“”</td>
<td>String</td>
</tr>
<tr>
<td>ID</td>
<td>2</td>
<td>Long</td>
</tr>
<tr>
<td>Owner</td>
<td>Object</td>
<td></td>
</tr>
</tbody>
</table>

Note that for either of the top hierarchical levels (components or anotherScript) we can expand the hierarchy to see the Application and Parent properties, to expand those properties and so on.

**See Also**

See the Debugger topic for an example of using the Variables pane, including using the Variables pane to change the value of variables during script execution. Other panes used with the debugger include:

- Call Stack
- Watches
View - Panes - ViewBots

**ViewBots** are one-line analytic instruments used to dynamically compute a statistical or comparative measure over a subset of records and are extremely useful for analyzing tables. They are one of the most popular functions in Manifold for experienced users. When a ViewBot is configured for a drawing’s table, it continues to function when the drawing has the focus. For example, if we configure a ViewBot to show the average of a given field for selected items, it will continue reporting the average of the selection whether or not we select items from a drawing’s table or by using mouse selection commands to select objects in the drawing.

Turn on the ViewBots pane with a **SHIFT-ALT-B** ("B" for "ViewBot") or with a View - Panes - ViewBots command.

The ViewBots above, for example, report on those operations they have been told to conduct on the table in view. ViewBots for table windows have many functions.

**To Create a ViewBot**

1. Open the ViewBots pane and press the **New ViewBot** button. A new ViewBot line will be added to the ViewBots pane.
2. Double click onto the saved ViewBot fields to choose analytic functions and the scope of the ViewBot.
3. Press the **Refresh** button to update the results in that ViewBot if the ViewBot autorefresh option is not on in Tools - Options.

**To Make a Selection using a ViewBot**

1. Click on the target component to make it active. Open the ViewBots pane.
2. Click on the desired ViewBot to highlight it.
3. Click on one of the ViewBot pane’s selection command buttons to combine the object set picked out by the ViewBot with any selection that is currently made. For example, clicking on the **Subtract from Selection** button will take the set of objects picked out by the ViewBot and will subtract the objects it contains from the current selection.

**To Delete a ViewBot**

1. Click on the desired saved ViewBot to highlight it.
2. Press the **Delete ViewBot** button.

**To Edit a ViewBot**

1. Double-click on the ViewBot to be edited, or
2. Click on the ViewBot to highlight it and press the **F2** key.

**ViewBots Pane Toolbar Commands**
Use the New ViewBot and Delete ViewBot buttons to create and to remove ViewBots from the pane. Use Refresh and Refresh All to update the calculation in the highlighted ViewBot or in all ViewBots.

Five selection command buttons are arrayed at the top of the ViewBots pane. These commands apply to whatever set of objects is picked out by the highlighted ViewBot. If we click on a ViewBot to highlight it and then press one of these buttons, the object set picked out by that ViewBot will be combined with whatever is the current selection in the image. The selection commands allow us to use object picked out by a View ViewBot to replace the selection, to add to the selection, to subtract from the selection, to invert with the selection and to intersect the selection. For example, if we have a ViewBot that reports the number of objects for which the Elevation is greater than 100, the set of all objects picked out by this ViewBot will be combined with the existing selection with the selection command buttons.

- Move Up - Move the active layer up one position in the layer stack.
- Move Down - Move the active layer down one position in the layer stack.
- Replace Selection - Replace the existing selection with the objects picked out by the highlighted ViewBot.
- Add to Selection - Add the objects picked out by the highlighted ViewBot to the existing selection.
- Subtract from Selection - Subtract the objects picked out by the highlighted ViewBot from the existing selection.
- Invert with Selection - Deselect what is in the objects set picked out by the highlighted ViewBot that was already in the existing selection and otherwise add the ViewBot’s object set.
- Intersect Selection - Select only the region of overlap between the highlighted ViewBot’s objects and what is in the existing selection. If there is no overlap, nothing will be selected.
- Refresh - Update the highlighted ViewBot.
- Refresh All - Update all ViewBots in the pane.
- New ViewBot - Create a new ViewBot line in the ViewBots list.
- Delete ViewBot - Delete the highlighted ViewBot.

Add ViewBot Dialog Controls

- Scope - The objects upon which the ViewBot operates: all objects (or records, pixels, etc), the selection, or the name of any saved selection.
- Column - The name of the field to use. Optional for some operations. For example, Number of Areas or Center X don’t require a field.
- Operation - The ViewBot operator to be employed.
- Argument - Arguments, if required for the operator.
- Ignore Case - Treat upper and lower case letters the same in strings. Used with text operators.
Ignore leading and trailing whitespace

Ignore any space or tab characters in strings. Used with text operators. "This" would be treated the same as "This".

Ignore interior whitespace

Ignore any space or tab characters within strings. Used with text operators. "Th is" would be treated the same as "This".

Caption

String to display for ViewBot results. Can include escape sequences in square brackets:

- [Argument] - Replace with argument in use.
- [Column] - Data field in use.
- [Operation] - Replace with the operation in use.
- [Scope] - Replace with scope in use, such as the name of a saved selection.
- [Scope Size] - Replace with the number of items in the scope.
- [Time] - Estimated time to compute the ViewBot. Normally used when Autorefresh is turned off to warn users of computationally intensive ViewBots that should not be casually refreshed.
- [Value] - Repeats the value computed and displayed in the Value column. Normally redundant.

The default Caption string of [Column] [Operation] [Argument] in [Scope]

creates a reasonably comprehensible sentence for most ViewBots.

Autorefresh viewbot

Command ViewBot to self-refresh on any changes in the subject component. If this value is not checked, a Refresh or Refresh All command in the ViewBots pane will be required to update the value of the ViewBot on any changes.

Selection and ViewBots

Some ViewBots such as the maximum value of a column allow selection from the ViewBot and some (such as the average value of a column) do not.

See Also

See the Selection topic for examples of Replace, Add, Subtract, Invert and Intersect selection command usage.

See the ViewBots topic for examples of ViewBot usage, including use from drawings and use from tables.

ViewBot Operators for lists of operators that may be used within ViewBots.
View - Panes - Views

The Views pane saves specific views into a window at a given location and zoom level under a name. Note that when a component is projected, saving a view at a given zoom level is really saving a view at a specified scale, the scale at which the view is displayed.

We can pan and zoom into a map of Switzerland, for example, so that Geneva is centered in the map and then save that view under the name Geneva. At any time we can jump directly to that view by clicking on the Geneva entry in the Views pane and choosing Apply View.

- **Apply View** - Go to this view.
- **Add New View** - Add the current view to the views pane.
- **Delete** - Delete this view from the views pane.

**To Save a View**

1. Pan and zoom the window to the desired view.
2. Open the views pane if it is not open (use the ALT - SHIFT - V keyboard shortcut if desired).
3. Press the Add New View button to add the view.
4. Click into the view entry to rename it as desired.

**To Go To a View**

1. Open the views pane if it is not open.
2. Click on the view desired to highlight it.
3. Press the Apply View button to go to that view.

If a component for which views have been saved is re-projected, Manifold will attempt to re-compute the extents and location of any defined views into the new projection. However, this will yield reasonably similar views only if the new projection is not too dissimilar to the original projection in which views were defined.

**The Startup View**

By default, the name Startup when given to a view will cause that view to be used whenever the component is opened. In the Tools - Options - User Interface dialog the Save last used view in component, if checked, causes the view called Startup to be used. This option is checked by default. If no view called Startup has been created for that component, then if this option is checked the component will be panned and zoomed to whatever was the last view when the component was last opened.

If the option is not checked, then on opening the component will be zoomed to fit the display window.

For example, if we would like a map to be panned and zoomed to a particular view whenever it is opened, we would open the map, pan and zoom it to the desired view, in the views pane press the Add New View button to add a view and then finally we would click into the new view entry and name that view Startup. Thereafter, whenever the map is opened it will automatically open to that view so long as the Save last used view in component option is checked in the Tools – Option – User Interface dialog.
The **Startup** view also works when programming IMS. Creating an instance of the MapServer object with an empty state string (for example, for the first time in a session for a particular user) automatically goes to the view named **Startup** (case insensitive) if such a view exists.

**Tech Tip**

The views pane works with print layouts windows as well. This is a handy way of navigating complex layouts to zoom in and zoom out onto different features that are frequently used: simply save the desired views in the views pane for quick navigation between views.

The views pane also works with Terrains to save a view of 3D terrain from a given position.
View - Panes - Watches

**Watches** are one-line dynamic instruments that watch and report upon values of variables in scripts or other process parameters. They are used to debug scripts by signaling watched-for conditions.

The **Watches** pane is enabled when the Manifold Debugger is installed.

**Watches** are user-defined expressions that dynamically report on the behavior of a variable or expression. Watch expressions may be a variable, a property, a function call or any valid expression (using arithmetic, for example) within the scripting language in use.

Watches are automatically updated during debugger commands. They will display the value of an expression within a given context and are a great way of monitoring what's going on in the script. In earlier days, one would have to insert short printing statements to print out different expressions as the code executed. In modern times, we can just use a watch to report those values "on the fly" within the Watches pane.

- **New Watch**: Create a new watch. Double-clicking onto the next blank line in the Watches pane is a shortcut for creating a new watch. Double-clicking onto an existing watch is a shortcut for editing it.
- **Delete Watch**: Deletes the highlighted watch from the pane. Highlighting a watch and pressing the **Delete** key will delete it.
- **Show Type**: Press in to show the type of the variable.
- **Show Common Properties**: Press in to show the Application and Parent properties that are common to all Manifold objects. Also controls the display of the **Item** and **_NewEnum** properties normally found in collections.

Double-clicking a watch or highlighting it and pressing the **F2** key will edit its value.

**Example**

Suppose we've stopped execution in a script called **A** at a breakpoint where another script is run.

```vbnet
Sub Main
    Set components = Application.ActiveDocument.ComponentSet
    Set anotherScript = components(components.ItemByName("B"))
    anotherScript.Run
End Sub
```

We've added two new watches to the pane. One watch was created using the expression "**components**" and the other was created using the expression "**components.Count + 1**".

With both **Show** buttons pressed in, the Watches pane will report the following values:
The `components` watch has been expanded. We could further expand the hierarchy to see the `Application` and `Parent` properties, to expand those properties and so on. We can see that the `Count` property for the `components` object has a value of 2.

The `components.Count + 1` expression evaluates to a value of 3.

**See Also**

See the Debugger topic for an example of using the Watches pane. Other panes used with the debugger include:

- Call Stack
- Variables
View - Panes - World

The **World** pane shows the current center point of the active window on a simplified map of the world. In addition, the World pane is used to control Linked Views by specifying **Show Position Reticule** and **Track Position** settings for individual windows. Open the World pane by choosing **View - Panes - World** or by using **SHIFT-ALT-O** to toggle the pane on or off. The World pane may be docked underneath the project pane when using docked panes.

The World pane is context sensitive: it shows the current **Show Position Reticule** and **Track Position** settings for the active window. The World pane announces the active window in the title bar and marks the center point of the active window with a reticule in both the world pane and, if the **Show Position Reticule** option is checked in the active window as well. In the example above, we have opened the **Mexico** sample drawing in a window. The World pane therefore announces **Mexico** as the active drawing and positions the reticule over Mexico in the World pane’s minimap.

![World Pane](image)

**Show Position Reticule** - Show a reticule in this window that always marks the center position of the currently active window (if it falls within the view seen in this window). Push in to activate.

**Track Position** - Pan this window so the center position of the currently active window is centered in this window’s view. Push in to activate.

**(Coordinates Readout)** The bottom of the World pane reports the coordinates of the center point of the active window.

To instruct a window to show a reticule at the current position

1. Open a component in a window.
2. Open the World pane and push in **Show Position Reticule**.

If any other window is now opened (even a second copy of the component opened above) the center point of the current position of the active window will be shown with a reticule.

To instruct a window to track the current position

1. Open a component in a window.
2. Open the World pane and push in **Track Position**.

If any other window is now opened (even a second copy of the component opened above) the tracking component will be panned so that it always shows the current position of the active window centered in view.

See the Linked Views topic for examples.
Properties

View - Properties

The Properties dialog always shows properties for the active window or the active component within a window. The dialog has three main forms depending on the component involved.

Drawings, Images, Labels, Maps and Surfaces

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The name of the component. Cannot be changed in this dialog.</td>
</tr>
<tr>
<td>Description</td>
<td>Any text description desired.</td>
</tr>
<tr>
<td>Status info</td>
<td>Shows Local or Linked or Shared.</td>
</tr>
<tr>
<td>Precision</td>
<td>Sets the location precision parameter. See the View - Properties - Precision topic.</td>
</tr>
<tr>
<td>Projection</td>
<td>Projection in use. See Edit - Change Projection and Edit - Assign Projection topics to change the projection. The Properties dialog shortens the name of the coordinate system (projection) in use. To see the full definition of the coordinate system, use the Projection dialog.</td>
</tr>
<tr>
<td>Zooms</td>
<td>Appears for components other than maps. Allows specification of zoom ranges, which are the zoom scales over which the component will appear or disappear when seen in maps. Click the [...] button to the right of the Zooms caption to specify zoom ranges.</td>
</tr>
<tr>
<td>Use custom background color</td>
<td>Background color for all components is set by default in the Tools - Options dialog. Each individual component can have its background color set to a custom color that overrides the system default.</td>
</tr>
</tbody>
</table>

The Link / Share dialog accessed from the [...] browse button for linked or shared components will provide a summary of the link or share properties. This will show the status of a shared component and whether or not changes in a linked component will propagate back to the data source.

Queries

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The name of the component. Cannot be changed in this dialog.</td>
</tr>
<tr>
<td>Description</td>
<td>Any text description desired.</td>
</tr>
<tr>
<td>Status info</td>
<td>Shows Local or Shared, with additional status information if Shared showing the Enterprise server involved and whether the component is cached or not cached. Status information other than Local is available only if Enterprise Edition is being used with components linked in from Enterprise servers.</td>
</tr>
<tr>
<td>Use ANSI-compatible Syntax</td>
<td>If checked, specifies usage of ANSI syntax in the SQL query. See the Queries and the SQL in Manifold System topics.</td>
</tr>
</tbody>
</table>
Use cache

If not checked (default), will forcefully re-compute the query each time it is run. Check to caches results of table queries, such as SELECT and TRANSFORM, between invocations. See the Queries topic.

All Other Components

Name
The name of the component. Cannot be changed in this dialog.

Description
Any text description desired.

[Status info]
Shows Local or Shared, with additional status information if Shared showing the Enterprise server involved and whether the component is cached or not cached. Status information other than Local is available only if Enterprise Edition is being used with components linked in from Enterprise servers.

Use custom background color
Background color for all components is set by default in the Tools - Options dialog. Each individual component can have its background color set to a custom color that overrides the system default.

Manifold SQL and ANSI Syntax Option

Manifold’s SQL (used in Queries) is designed to be compatible with Microsoft SQL as implemented in the Microsoft “Jet” database engine. Manifold SQL is intended to be a superset of Jet SQL with features supporting ANSI SQL as well. SQL syntax as documented within the Microsoft Jet SQL Reference for Access 2000 is a practical approach to Manifold SQL.

In addition to being Microsoft-like, Manifold’s SQL engine may be switched into an alternate mode where it parses SQL queries using pure ANSI syntax. To do so, open the query and in the View - Properties dialog check the Use ANSI-compatible syntax box.

When the ANSI option is turned off (the default), line comments start after an apostrophe (’) or double dashes (--) and the names of tables and columns can be enclosed in square brackets ([ ]) or backward apostrophes (‘

strings are enclosed in quotes (“) and times are enclosed in quotes (“) or hashes (#).

When the ANSI option is turned on, line comments start after double dashes (---), the names of tables and columns can be enclosed in square brackets ([ ]) or quotes (“) or backward apostrophes (‘), strings are enclosed in apostrophes (‘) and times are enclosed in apostrophes (‘) or hashes (#).

Clearly, the safest course given that users might change the settings is to use double dashes (---) to mark line comments and to enclose the names of tables and columns in square brackets ([ ]) and to enclose times with hashes (#). The main difference between having the ANSI option off or on is that with it off strings are enclosed in quotes (“) and with the ANSI option on strings are enclosed in apostrophes (‘).
View - Properties - Link / Share Status

The **Link / Share Status** dialog is reached from the **View - Properties** dialog by clicking on the [...] browse button in the **Status** line. This dialog reports on the connection status of linked components as well upon the share status of components shared from Enterprise servers using Enterprise Edition installations of Manifold System. It allows us to alter the **Edit mode** that applies to linked components used to resolve any editing conflicts that arise from concurrent multi-user editing.

**Controls**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linked from</td>
<td>The data source from which the component is linked.</td>
</tr>
<tr>
<td>Using</td>
<td>The connection technology in use.</td>
</tr>
<tr>
<td>Rowset</td>
<td>The rowset within the table, that is, the table or query from which the component is linked.</td>
</tr>
<tr>
<td>(propagation status)</td>
<td>A status line reporting whether or not changes made to the component propagate back to the data source.</td>
</tr>
<tr>
<td>Edit mode</td>
<td>Specifies how editing conflicts arising from multi-user editing of geometry will be handled.</td>
</tr>
<tr>
<td></td>
<td><strong>Overwrite changes made by others</strong> - Do not attempt to detect any changes made to drawing objects by others. Immediately apply any local changes made to drawing objects, overwriting any simultaneous changes made by others. This is the only choice if either a version column has not been specified or if Enterprise edition or above is not used.</td>
</tr>
<tr>
<td></td>
<td><strong>Review changes made by others</strong> - Detect any changes made to drawing objects by others and resolve editing conflicts using the Review pane. This is the default when using Enterprise edition if a version column has been specified.</td>
</tr>
<tr>
<td></td>
<td>The <strong>Edit mode</strong> applies to geometry editing only: as is the case with any table attribute editing, any edits of non-geometry fields in a linked drawing's table will always be immediately applied, overwriting any changes made by another user.</td>
</tr>
<tr>
<td>Shared on</td>
<td>For components shared with Enterprise edition, the name of the Enterprise server upon which the component is shared.</td>
</tr>
<tr>
<td>Using</td>
<td>The connection technology used to connect to the Enterprise server.</td>
</tr>
<tr>
<td>Cache component data</td>
<td>Checked by default for shared components. Cache data for the component on the local machine. A good idea for performance and to allow work to continue if disconnected from the Enterprise server.</td>
</tr>
<tr>
<td>locally</td>
<td></td>
</tr>
</tbody>
</table>

**See Also**

- Multi-User Editing of Linked Drawings
- Enterprise Edition
**View - Properties - Precision**

The View - Properties - Precision dialog sets a location precision distance for drawings and maps that defines the radius of action of many commands and operators, including SQL. For example, some drawing transform toolbar operators like Normalize Topology will snap a line end to a point if the line end is closer to the point than the distance given in the Precision dialog. Two points falling within the location precision distance will be deemed coincident. The precision parameter is also important when using automated tools such as Attach and the Topology Factory tool.

This is a critically important parameter that must be understood when using Normalize Topology or other advanced commands. It is necessary because the digital representation of an analog world requires some "quantum" level to determine when a given distance is attained or not attained. For example, if we search for all points within 2 miles of another point, in the digital world there is never exactly a point at the 2 mile distance. It might be at 1.9999999999 miles or 2.0000000001 miles but it won't be "exactly" at 2 miles. The location precision parameter allows us to specify with what precision a distance is truly considered at "2" miles.

The precision parameter is also important for geometric comparisons such as determining if two points are coincident. If they fall within the distance specified by the precision parameter, two locations are considered to be the same.

By default the precision parameter is set to one millionth of whatever measurement unit is used in the drawing's coordinate system. In the case of latitude / longitude drawings this will be one millionth of a degree (about ten centimeters at the Equator). In the case of projected drawings that are meter based, such as the default Orthographic projection, the precision parameter will be one millionth of a meter by default.

Drawings that are imported using less accurate can be "quantized" into a desired precision setting by running Normalize Topology using a new precision setting as follows:

**Quantizing Drawings into a Desired Precision**

1. Open the drawing.
2. Using the tracker tool zoom into the drawing measure the distance between two points that are extremely close to each other but which should be treated as separate locations. These two points ideally should be the very closest that two points should be from each other before they are considered the same.
3. Divide the distance measured above by 10. That is probably a reasonably safe value to use as a sufficiently small precision value that any points closer together than that should be considered the same location.
4. Open the View - Properties - Precision dialog and specify this precision value as the Location Precision value to use. Press OK.
5. Save the .map file.
6. Run Normalize Topology on all objects. This will slightly alter any common area boundaries and other anomalies that have slight overlaps, holes, undershoots, overshoots, etc, within the new Location Precision value so that borders are exactly coincident, overshoots do not occur, points are coincident with the ends of lines, etc.

**Initial Precision Settings**

When we create a drawing, its location precision is set to 0.000001 degrees or 0.000001 meters depending on whether the drawing is in Latitude / Longitude or not. This default value and the default choice of degrees or meters as units of measure stays unchanged unless we modify it by using View - Properties - Precision, or re-project a projected drawing into Latitude / Longitude or vice versa.

**Tech Tips**

The importance of location precision parameter can be seen by importing the mexico_eg sample drawing, opening it, setting location precision to 0.1 and then running Normalize Topology. This achieves, in effect, a "generalize" operation on the drawing that will simplify it (while maintaining topological relationships) to using many fewer coordinates to define the same objects. This is a long computation but well worth the time to see the effect.
Within the Manifold programming and mathematics team the location precision parameter is called the \textit{epsilon}. Some documentation and notes within sample source code may refer to "the epsilon" used for a given action. This is the location precision distance referred to by a more mathematical name.

Because Manifold utilizes double-precision floating point numbers for internal calculations, the ultimate limit of precision with Manifold is approximately one two hundred fifty millionth of a meter at the Equator, accuracy sufficient to "map" the surface of a bacterium or even a virus. This extraordinarily high internal accuracy within Manifold, millions of times more accurate than even survey-grade GIS data, assures users that internal Manifold operations will always preserve whatever accuracy is available within the data set in use. Whatever the data set in use, Manifold’s internal machinery will always be far more accurate.
View - Properties - Zooms

The **Zooms** dialog called from the **View - Properties** dialog for a component has two functions:

- Specifying the zoom ranges over which that component will be displayed when it appears in a map. Zoom ranges only apply when the component is displayed in a map window.
- For vector components (drawings, labels, profiles and themes) specifying the **Render zoom** parameter. The **Render zoom** parameter always applies whenever the component is open either in its own window or within a map window.

Zoom ranges and **Render zoom** values apply only to **projected** components. They have no effect on components that are in Latitude/Longitude unprojected form.

For details on the use of zoom ranges, see the Zoom Ranges topic.

**Controls**

**Minimum Zoom**

The lowest zoom at which the component will be visible in a map window. At zooms lower than this (zoomed further in) the component will not be visible. Leave blank to display at all zooms. Either choose a pre-set value from the pull-down menu or enter a desired custom value.

**Maximum Zoom**

The highest zoom at which the component will be visible in a map window. At zooms higher than this (zoomed further out) the component will not be visible. Leave blank to display at all zooms. Either choose a pre-set value from the pull-down menu or enter a desired custom value.

**Render zoom**

If blank, render labels and point, line and area styles using the **size** parameter in the format toolbar to constantly maintain the same size in printer's points units even if the view is zoomed in or out. If a value is entered, change the size of labels and point, line and area styles to make them larger or smaller in proportion as the view is zoomed in or out. Either choose a pre-set value from the pull-down menu or enter a desired custom value.

**Current Zoom**

The current zoom of the component when the **View - Properties - Zooms** dialog was launched. This provides a reference value from which we may get our bearings when deciding what values for zoom ranges or **Render zoom** we wish to use.

**Render Zoom**

The default operation of Manifold within component and map windows is to provide a "dimensionless" rendering of point, line and area styles so that no matter what zoom is selected the styles will appear using the **size** specified in the format toolbar. For example, if a given point style is specified to be 6 printer's points in size, it will always be shown onscreen exactly 6 points in size. If we zoom in or out the point style will always be shown 6 points in size.

Manifold provides this behavior because vector components such as drawings can be zoomed in or out as much as we like. Normally when, say, many points are clustered together we would like them to continue to be the same size as we zoom further in so that many tightly bunched points can be resolved into visibly separate points.

In the default operation of Manifold, if we wish to have a "paper space" relationship between zooming in and out and the size of symbols such as points, we can use a print layout to create a composition based upon the fixed size of a sheet of paper. When we zoom in and out of a layout, the points and other vector symbols we see will get larger and smaller just as if we were looking at a printed sheet of paper more closely with a magnifying lens or from farther away.

In specialized applications we may want to force component windows to have the same behavior as a print layout display, where the size of labels, points and other symbols will change if we zoom in or out. We can force such
behavior by specifying a **Render zoom** value. The **Render zoom** value specifies the scale at which styles will be drawn exactly at the **size** specified in the format toolbar. Zooming farther in, for lower zoom values, will cause the styles to be drawn proportionately larger in size. Zooming farther out, for higher zoom values, will cause the styles to be drawn proportionately smaller in size.

Inexperienced users are often confused between the free-scale world of vector component windows and map windows and the fixed-scale world of a print layout. The normal way of creating displays in which labels and symbols get larger and smaller as we zoom in or out is to use a **print layout**. Do not use the **Render zoom** parameter as a sloppy way to get a print layout effect within, say, a drawing or a map window just because the use of print layouts or scaling within print layouts is not understood. Better to study the documentation for print layouts and get those concepts clear so that the right tool may be used.

Keep in mind that one of the significant benefits of component windows like drawings and maps is exactly that they do not mimic the behavior of paper maps. For most purposes the default behavior of component windows provides benefits that would be obscured if symbols and labels were to get larger and smaller as we zoomed in. It is natural for beginners to want to approach GIS using the mental paradigms of paper maps, but it would be a mistake if such an approach denied users the signal benefits of a computerized workspace.

On the other hand, there are times when using a **Render zoom** value is the right choice. For example, suppose we want to publish a map component to the web using the Internet Map Server and to keep things conceptually simple for some users we want the resultant images to behave like a print layout when they are viewed. In that case, we can use **Render zoom** to make the image appear like a paper space image so that as visitors zoom farther into the image labels and point styles will appear larger.

**Example**

Consider a map of congressional districts in the US that includes as a layer a drawing containing points for major cities in the US.

We open the map and zoom in to whatever zoom we think will be our “standard” zoom, a typical view.
We then invoke View - Properties - Zooms and note the Current zoom value. It is approximately one to twenty six million.

We can enter 26000000 into the Render zoom box to make this the standard render zoom value.

If we now zoom farther into the drawing the point styles will increase in size. If we zoom twice as far into the drawing the points will be twice as large.

The farther we zoom into the drawing, the larger the point sizes will be, just as if we were viewing a printed sheet of paper through a magnifying lens.

Note that the display computed at the Render zoom value will be the same display, except for increase or decrease in size of symbols and labels, at higher or lower zooms. For example, zooming farther into the map does not allow any overlapping points to be resolved into separate points. This effect can be seen with labels.
Suppose we have a labels layer for the names of major cities in our map and we also set the Render zoom value for that labels layer to 1:26000000 just like the city points layer. Shown above is the map is zoomed to the same starting view, an approximately 1:26000000 zoom. The labels are displayed based upon computations for clipping overlapping labels at that zoom level.

As we zoom farther into the map, the labels simply get larger. No labels are added or deleted.

Just as the case with point styles, the farther we zoom into the map the proportionately larger the labels will become.

See Also

Zoom Ranges
Turning Layers Off/On by Zoom
Format Toolbar
View - Properties - Zooms

Drawing Menu
Drawing - Area of Interest
Appears when the focus is on a drawing linked from a DBMS that allows area of interest specification (drawings linked from spatial DBMS or DBMS enabled with a spatial index). Allows alteration of the area of interest specified when creating the drawing. Changing the area of interest will result in immediate updating of the drawing.

Controls

<table>
<thead>
<tr>
<th>Control</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use all objects</td>
<td>Use all objects in the source database table for the drawing.</td>
</tr>
<tr>
<td>Use objects in the following area</td>
<td>Use only those objects for the drawing that fall within the given extents.</td>
</tr>
<tr>
<td>Require objects to be completely within area</td>
<td>Use objects only if they fall entirely within the given extents.</td>
</tr>
</tbody>
</table>

See Also

Linked Drawings
Geometry in Tables
Spatial DBMS
Spatial DBMS Facilities
Drawing - Color

The Color dialog makes it easy to automatically apply a thematic format to a drawing that colors each adjacent area using a different color.

When applied to a drawing that contains adjacent areas, the Color dialog will add an integer field called Color to the table. (If a field exists that is already named Color the system will create the new field using a name like Color2). The drawing is then examined using Manifold’s internal graph theoretic algorithms to assign a small number to each area in the Color field so that no two adjacent areas have the same value in their Color field.

The Color field is then used in a thematic format to specify both foreground and background color. The colors used are made slightly darker in the foreground color so that the borders of area styles will be distinct.

- **Scope**: [All Objects] by default, but the selection or any saved selection can be used.
- **Save to**: <New Column> is the default, to create a field called Color or similar in naming sequence to which color codes separating adjacent areas by color are written. Existing fields may also be used.
- **Assume areas do not overlap**: Allows a faster spatial computation if checked. If not checked, Manifold will launch an exhaustive computation to resolve possible overlap ambiguities and take much longer.
- **Modify formatting**: If checked (the default) will modify formatting so that adjacent areas are colored differently. If not checked, correct color codes will still be computed and saved to the table. Formatting can then be changed manually, as desired.

**Example**

We've imported a drawing of 2073 census bureau blocks in the San Francisco area. The screen shots below show a detail of the map zoomed into the city of San Francisco itself.

![Map of San Francisco census bureau blocks](image)

Seen default colors the areas are all gray. We would like to color them so that no two adjacent areas share the same color.

To do so, we launch the Color dialog from the Drawing menu. We use the default options with <New Column> in the Save to box and press OK.
After time for processing (there are over 2000 areas in the example) the drawing appears with new colors as seen above. No two adjacent areas share the same color.

The **Color dialog** creates a thematic format for both the **background** color as well as the **foreground** color. The foreground color theme is a darker version of the background color with the expectation that an area style using foreground color for apparent boundary is being used.

<table>
<thead>
<tr>
<th>Area</th>
<th>NAME</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.836</td>
<td>BG 2</td>
<td>0</td>
</tr>
<tr>
<td>4.575</td>
<td>BG 2</td>
<td>0</td>
</tr>
<tr>
<td>10.947</td>
<td>BG 9</td>
<td>1</td>
</tr>
<tr>
<td>3.781</td>
<td>BG 3</td>
<td>0</td>
</tr>
<tr>
<td>0.961</td>
<td>BG 3</td>
<td>1</td>
</tr>
<tr>
<td>13.042</td>
<td>BG 3</td>
<td>0</td>
</tr>
</tbody>
</table>

If we open the drawing's table we can see that a new field, called **Color**, has been added to the table. This field contains the value 0, 1, 2, 3, or 4. The field is then used in the thematic format to color the drawing.

It's easy to change the color scheme after applying the color dialog. Simply click on the color well to be changed for areas and choose **theme** and then apply one of the preset color schemes. The color applied by the color dialog is simply a thematic format using the **Color** field with a **Unique Values** method.
If we apply the Savannah thematic palette preset and then manually change a few colors (by clicking into the color wells and choosing the color we want) we get the results seen above. To create this illustration we changed the foreground color for areas to a dark gray.

If we apply the Greens color preset we get the illustration above. If we don't like any of the color choices in the presets we can double-click into the color wells in the thematic format dialog and change them to whatever we like.

For example, we can double click into one of the color wells assigned by the Greens preset and change it to yellow.
The result will be a change in that color throughout the drawing.

The above examples show alteration of the colors assigned by the color dialog. Using analogous methods in the thematic formatting dialog we could change area styles as well.

**Performance**

Theoretically, one can color a map using no more than four colors. The Color dialog engine will try to use only four colors when reasonably possible. However, the algorithm will use five colors to color the drawing in certain complex area relationships if it determines that solving the four-color problem will take an unreasonably long time. Solving the five-color problem requires much less computation time than finding a solution using only four colors.

The Color dialog works by creating an internal graph-theoretic model of the spatial adjacency relationships between areas. It then applies the Manifold coloring function to find a solution for the coloring of the graph and assigns color codes to areas. Because the algorithm is precise and correctly deals with sophisticated topological issues such as possibly branched areas, computation can be very intensive when many areas are involved. Don't expect to color over 3000 counties in the US in a matter of seconds.

Depending on the speed of the machine involved and the number of areas involved, the Color dialog can take a long time to do its job. Coloring 2073 census block areas in the San Francisco Bay area to create the map used in this example required 21 minutes on a 1 gigahertz Athlon running Windows 2000. On the same machine, the 38 areas representing provinces in our example Mexico map were colored in less than a second.

The census blocks were taken from boundary files downloaded from the Census Bureau's web site. The blocks for all California were imported and then all blocks except those in the area of interest were deleted.
**Drawing - Dissolve**

*Dissolve* is so named because it "dissolves away" or removes the borders between areas that have identical values in data attribute fields. Given a set of areas, it can automatically create new areas that are spatial combinations of areas having the same data attributes. It is available for drawings or drawing layers in maps. Dissolve can also dissolve away the "borders" between lines that are incident to each other. It can also "dissolve" away groups of points to create a centroid for points with a common field value.

Suppose we have a drawing of the US where each state has a region field.

<table>
<thead>
<tr>
<th>Name</th>
<th>Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alabama</td>
<td>East</td>
</tr>
<tr>
<td>Arizona</td>
<td>West</td>
</tr>
<tr>
<td>Arkansas</td>
<td>Central</td>
</tr>
<tr>
<td>California</td>
<td>West</td>
</tr>
<tr>
<td>Colorado</td>
<td>West</td>
</tr>
<tr>
<td>Connecticut</td>
<td>East</td>
</tr>
</tbody>
</table>

The value in the region field can be **East**, **West** or **Central**.

If we *Dissolve* all objects in the map using the *Region* field we combine all areas of like region value into one area. Thus, all the states with the value **West** in their *Region* field will be unioned into a single area.

**Dissolve Dialog Controls**

The *Drawing - Dissolve* dialog allows us to choose which areas are to be dissolved, what the guiding field should be and how the field values should be combined.
Dissolve

Dissolve: Choose [All Objects] or the name of a saved selection.

Using: Choose the field to use to guide the dissolve process.

Ignore Case
Combine text strings treating any variation in case as the same string.

Ignore leading and trailing whitespace
Combine text strings without regard to whitespace characters before or after the string.

Ignore internal whitespace
Combine text strings without regard to whitespace characters occurring within the string.

Aggregate Options

(blank) Do not transfer values for this field.

Average Transfer the average value that occurs in this field.

Count Transfer the total number of objects in this field.

Maximum Transfer the maximum value that occurs in this field.

Minimum Transfer the minimum value that occurs in this field.

Sample Randomly choose one of the values from the original object and place it in the combined object.

Sum Transfer the sum of values that occur in this field.

Different aggregate options will be used with different types of information. For example, suppose each state has an Area field that gives the area of the state in square kilometers (or more likely for the US, in square miles…). We would like the Area field for the combined area created by Dissolve to be the Sum of all the areas of the states that were unioned together.

Use with Lines

The classic example of using Dissolve with lines is the case of a drawing showing highways as lines. Quite often in such cases a highway will be composed of many line objects arranged end to end with each line having a data attribute that gives the highway name or number. At times we may wish to create a drawing that has a single line created from all lines that have "5," or "80," or some other number as their highway identifier. We can accomplish this task using Dissolve.
Suppose we have several lines with a **Name** field that contains the value 5 together with two other lines that have the values 18 and 80 in their **Name** fields. The illustration above shows one of the 5 lines selected so it is clear that the longer line sequence at the left of the illustration is composed of more than one line object. The illustration shows the drawing seen in a map with labels created from the lines.

We can run **Drawing - Dissolve** on the drawing using the settings above.

The result is the three line objects that had the same value 5 in their **Name** field have been replaced by a single object that has 5 in its **Name** field. The illustration shows labels drawn aligned to lines so more labels appear for each line object.

**Transfer Rules**

When objects are dissolved into one object, the new object always inherits the value of the field used to guide the dissolve no matter the setting of Transfer Rules for that field. All other fields will be merged together using the settings of transfer rules for those fields.

**Use with Points**

When used with points, Dissolve will find the centroid for all point sharing the same value in the designated field and will replace those points with a single point at the centroid.

Suppose we have a drawing with points. Each point has a field named **Name** with the value 3 or 6. The illustration above shows the drawing in a map with a labels layer above it that shows the value of the **Name** field for each point. The points with a value of 6 have been selected to make it clearer how they are arranged.
As before, we run **Drawing - Dissolve** using the **Name** field.

The result is that all of the "3" points are replaced with a single "3" point at their centroid and all of the "6" points are likewise replaced with a single point.

If we had copied the original drawing we could show it in map with the new points created by Dissolve. The illustration above shows the original drawing with reduced layer opacity so it appears faint and thus the two new points are more clearly visible.

How might Dissolve be used with points in a geographic situation? Drawings imported from some formats, such as VPF (VMAP) or SDTS may at times create drawing layers with points that are intended to be anchor points for labels in the native format. So, for example, we might import a VPF data set that ends up creating a scattering of points, all of which have "Gulf of Mexico" as their value. However, we might wish to have only one point so valued for each different instance of label ("Gulf of Mexico", "Atlantic Ocean," etc.). We can easily accomplish this with Dissolve, which will replace each collection of same-valued points with a single point at their centroid.

**Memory Limitations**

32-bit Windows operating systems cannot allocate more than 2 GB of memory per process. This puts an upper limit on system functioning whereby algorithms that need more than 1 GB of memory (even temporarily) cannot usually operate. For example, using **Dissolve** to dissolve millions of polygons with very large data attributes for each record can exceed the process limit. Research continues at manifold.net to work around this limit of 32-bit Windows systems. As a solution until either 64-bit Windows systems become generally available or until Manifold research finds a way to exceed the limit imposed by 32-bit systems, try running large algorithms in parts.
example, try running a large **Dissolve** on portions of a drawing and then do a **Dissolve** on the results of these runs.
**Drawing - Districts**

The **Drawing - Districts** command assigns areas in the drawing to different districts by writing a district code to a specified data field. Given a drawing that contains areas (usually, adjacent areas like a drawing of the countries of Europe, provinces in a given country, counties in a US state, etc.) the function assign each area to one of the given number of districts such that each district is reasonably compact and contiguous.

If you have acquired the optional Business Tools extension to Manifold System, use of the Districts (Visual) dialog is much preferred to the basic **Districts** dialog.

**Scope**  The set of areas to be assigned to districts.  **[All Objects]** by default, but can be set to the selection or any saved selection.

**Balance**  Field to be used to balance districts.  Districts will be created so each district (as much as possible) has the same total value of this field.

**Save to**  The column to be used to save the district assignment code.  **[New Column]** by default to create a new column called **District**, or a choice of any existing column.

**Threshold**  Areas within this distance will be considered adjacent.  The default value of 0 means that areas touching each other are considered "near" and that areas with no common points are "far" from each other.

**Districts**  The number of districts to which the areas should be assigned.  Choosing 4, for example, will cause each area in the drawing to be assigned to one of four different districts.

**Modify formatting**  Reformat colors of areas so that areas in each district are colored similarly.  Colors used are taken from the same palette used in the Color dialog.

Choosing 4 districts will cause the Districts command to create a new column called **District** and then place a value of 0, 1, 2 or 3 into that column for each area.  All areas with the value 0 will be grouped together next to each other in one district as will the areas with values 1, 2 and 3 respectively.  Redistricting can be accomplished by opening the drawing's table and changing the **District** code for any area that is to be reassigned to different district.

The thematic format created by default will use the **District** column with unique values to specify color.  To change the color scheme used, use thematic formatting to change the colors used to format using the **District** column.

The Districts command is intended to provide simple, initial districting.  For advanced districting allowing greater control and flexibility over the shape and placement of districts, the balancing criteria and other factors, please use the latest **Business Tools** extension for your version of Manifold System.

**Example**

We will create districts using the **mexico_eg** sample drawing (referred to here as "mexico" for brevity).
Seen in default formatting in a drawing window, the *mexico* drawing consists of areas for provinces, each of which has a variety of columns giving geographic and demographic information.

Choose **Drawing - Districts** and choose **SQKM** as the **Balance** field. This field gives the area of each province in square kilometers. Change the number of **Districts** to 4 and press **OK**.

The Districts command will create a new column called **Districts** and will assign each province in *mexico* to one of four different districts by placing the value 0, 1, 2 or 3 into that column for each province. Thematic formatting will then be applied using the **Districts** column as a guide for formatting. The objective of the Districts command is to assign each province into districts so that the sum of the total **SQKM** for each district is the same and so that all areas in each district are contiguous (within the threshold parameter distance). To achieve districts where each district has the same overall **SQKM** total can require some oddly shaped districts although the Districts command will try to keep districts as compact as possible.
We can see that in the case of an irregularly shaped country such as Mexico it is a challenge for the Districts solver to create compact districts that also balance the desired field. In particular, we can select one province that seems it could be swapped for a different province.

Opening the drawing's table we can see that the province of Zacatecas has an area of 73252 square kilometers. It has been assigned to district 2, the brown district. [We used Table - Columns to show only three columns in the table.]

We can click on another province in the green district that seems it could be swapped for Zacatecas.
We see in the table that the province of **Guerrero** has an area of **64281** square kilometers. It has been assigned to district **0**, the green district. Swapping this province for **Zacatecas** would reduce the total square kilometers in the brown district and increase the total square kilometers of the green district, but not by much. We might decide that it is a reasonable trade to keep the provinces more compact.

We can reassign the **Guerrero** province to district **2** by double clicking into the **Districts** cell for the **Guerrero** record and entering the value **2** as seen above. We can then change the **Districts** cell for the **Zacatecas** record to **0** (not illustrated).

We can then click on the drawing, click on the background color well in the formatting toolbar, click on **Thematic** and reapply the formatting to update the formatting of the provinces to using the new values just entered for **District** in the toolbar. The result as seen above is a more compact allocation of provinces into districts.

**Performance**

Districting is easy to do if a drawing fulfills certain limiting conditions, such as a guarantee that areas never overlap and are always perfectly adjacent. Since "real world" drawings often have imperfections, the Manifold districting algorithm uses a **Threshold** factor for determining distance between areas. The algorithm also uses a variety of computational geometry and network algorithms to deal with common imperfections in data sets. As a result, computation with a large number of areas could be slow with a large number of areas and a slow machine. To assess the performance of a particular machine with a given data set, begin by applying Districts to small data sets such as the sample **mexico_eg** drawing and scale up gradually.
Districts (Advanced)
Districts (Visual)
**Drawing - Districts (Advanced)**
See the Districts (Advanced) topic.

**See Also**
Districts (Visual)
**Business Tools**
Drawing - Districts (Visual)
See the Districts (Visual) topic.

See Also

Districts (Advanced)
Business Tools
Drawing - Drive-Time Zones
See the Drive-Time Zones topic.

See Also

Business Tools
Drawing - Minichart

Minicharts are small charts that hover above objects in a drawing. They display the relative values for each object of specified numeric data fields from tables associated with the drawing.

To Create Minicharts in a Drawing

1. Click open a drawing.
2. Choose Drawing - Minichart
3. In the Minichart dialog, use the Add Field button to add numeric fields that are to participate in the minichart.
4. Choose the display Style, Size and Frame color and other parameters. Press OK.

To Remove Minicharts from a Drawing

1. Open the drawing.
2. Choose Drawing - Minichart
3. In the Minicharts dialog click the Select None button to uncheck all fields. Press OK.

Minichart Dialog Controls

- **Select All** - Check all fields in the Chart pane.
- **Select None** - Uncheck all fields in the Chart pane.
- **Select Inverse** - Uncheck all checked fields and check all unchecked fields. A fast way to show all but one field: click Select None, check the one column not desired and then click Select Inverse.
- **Move to Top** - Move highlighted field to the top of the list.
- **Move Up** - Move highlighted field up one position in the list.
- **Move Down** - Move highlighted field down one position in the list.
- **Move to Bottom** - Move highlighted field to the bottom of the list.

**Chart**
A list of columns to chart with colors to use for each. Check a column to display it in the minichart. Double-click into a color well to change the color.

**Style**
Choose from common types of simple charts.

**Frame color**
Color of the border lines used in chart styles.

**Offset point charts by**
When a minichart is created based on data from point objects, specifies the number of printer's points in the X and Y directions to offset it from the point. By default, the offset is 6 points in both X and Y. Increase these values to position the minichart farther away from the point. Offsets must be in the range from -100 to 100 points.

**Size**
Size of the minichart in points.
Resolve overlaps between charts

Checked by default. Do not show minicharts that overlap each other. Uncheck to show all minicharts.

Spacing

Minimum spacing between minicharts in points before minicharts will be hidden to resolve overlaps.

Use with Restraint

Minicharts are visually pretty and fun to use but they are not always effective methods of conveying information about data attributes. The problem is that many small, complex graphical elements sometimes can cause more confusion than enlightenment.
Drawing - Optimal Route
See the Optimal Route topic.

See Also
Optimal Route (Visual)
Business Tools
Drawing - Optimal Route (Visual)
See the Optimal Route (Visual) topic.

See Also

Optimal Route
Business Tools
**Drawing - Orthogonalize**

The **Orthogonalize** command moves coordinates to orthogonal grid positions at the specified X and Y grid steps. The command is used mainly to trim insignificant digits from coordinates. For example, suppose we are working with a projected data set expressed in meter-based coordinates and we know that our data set is only accurate within 10 meters. In such cases, there is only an illusion of accuracy when using coordinates such as 34592.490593845 meters. It may as well be 34590 meters. If every coordinate were moved to the nearest ten-meter coordinate position, we would achieve the effect of rounding all coordinates. **Orthogonalize** command performs exactly this function.

Suppose we start with a drawing of Durango province in Mexico using Orthographic projection.

After applying the **Orthogonalize** command with X and Y steps of **25000** meters the province takes on a distinctly stair-step appearance as coordinates are forced to the nearest 25000-meter grid position. The **Orthogonalize** command is not normally used in cases where the original coordinates appear in much finer resolution than the steps specified, so some regions have been collapsed into topologically redundant appendages. These can be fixed by running the Normalize Topology transform toolbar operator.

The **Normalize topology** operator cleans up the topology.
Drawing - Relink / Unlink

The Relink and Unlink commands are enabled when a linked drawing has the focus.

- **Relink** - Used to reconnect a linked drawing to a data source. This command is also available within the drawing's context menu in the project pane. If a linked drawing is disconnected from a data source this command is used to reconnect it. For example, one might create a .map file that contains a linked drawing that is linked to an Access .mdb file. If the .map file and the .mdb file are moved to another machine the link might not continue to work. In that case one can use the Relink command to reconnect the drawing in the project to the .mdb file.

- **Unlink** - Unlink a linked drawing to create a drawing in the project. All data for the drawing is copied from the data source and embedded within the project as would be the case for an ordinary, unlinked drawing.

See Also

Linked Drawings
**Drawing - Segmentize**

The *Segmentize* command adds redundant coordinates to objects. It is normally used to make ready for projection very large, simple objects that would otherwise suffer distortion. See the Segmentization topic for discussion of segmentation and projections. The *Segmentize* menu command applies additional coordinates so there is no straight segment of an object that has greater than the given *Distance* between coordinates.

Suppose we have an area that is a simplified rendering of the province of Durango in Mexico, seen in Orthographic projection in a drawing. The province is approximately 440 kilometers wide. The simplified rendition was created by applying Simplify with a *Distance* setting of 25000 meters.

If we click on the area to select it as the primary selected object for editing, edit handles appear at each coordinate that define the area.

Zooming in to the Eastern side of the province we can see that the area consists of straight line segments between the coordinates that define the area.

If we apply *Segmentize* with a *Distance* setting of 10000 meters, Manifold will place redundant coordinates along the straight segments every 10,000 meters. Additional edit handles will appear for each coordinate.

**Tech Tip**
In some circumstances, Segmentize and Simplify are inverse functions. For example, if we applied Simplify again with a setting of 25000 after the Segmentize command shown above the redundant coordinates would be removed.

See Also

Segmentization
Simplify
**Drawing - Send Email**
See the Send Email topic.

**See Also**

Business Tools
The **Simplify** command reduces the number of coordinates that define an object. Given a **scope** of action ([All Objects] or a selection) and a **Distance**, the **Simplify** command will reduce the number of coordinates such that the coordinates that define the object are approximately the given distance apart. The **Remove small branches** option will automatically remove branches in multi-branched objects (such as areas with islands or holes) that are smaller than the given distance.

In the example above, the province of Durango from the sample Mexico map has been simplified using **Simplify**. The drawing was first projected into Orthographic projection so that meter-based distances could be used.

The **Simplify** dialog was run with a **Distance** of **25000** meters.

**Comparison to Normalize Topology**

The **Simplify** command is similar to the Normalize Topology transform toolbar operator. However, the **Simplify** command is aimed at simplification on a per-object basis while the **Normalize Topology** operator considers relationships between objects when simplifying them. We can see the difference by running both commands on a drawing of Mexico.

We begin with a drawing of Mexico that's been projected into **Orthographic** projection.
Running **Simplify** with a distance setting of 25000 meters results in the simplification of areas. However, each area is simplified without considering any relationships with adjacent areas. This provides an optimal simplification when each area is considered by itself but also results in overlaps and gaps between areas.

If instead of using **Simplify** we used **Normalize Topology**, we would first use **View - Properties - Precision** to set the **Location Precision** to 25000 meters and then we would run **Normalize Topology**.

**Normalize Topology** takes a lot longer to run than **Simplify**, but the result in the end is free of overlaps and gaps. Each area has been adjusted both on the basis of capturing its own shape as well as to match its neighbors. However, it could be said that the resulting shape for each individual object (without regard to neighboring areas) is not as "optimal" a simplification as is done by **Simplify**.

When simplifying a single object it is probably best to use **Simplify** since the result is obtained much faster. When simplifying many objects for which adjacency must be maintained it is best to use **Normalize Topology**.

**See Also**

**Normalize Topology**

**Segmentize**
**Drawing - Spatial Overlay**

Spatial overlays are a set of methods for transferring data between objects in drawings based on their spatial relationships to each other.

For example, suppose we have a drawing that shows states as areas and we also have a set of points showing the locations of cities. Suppose that each city record has a field called *Population* that gives the population in that city, but that we have no values for the population of each state. We could use spatial overlays to automatically add up the values of the populations for all cities in each state and place that combined value into the *Population* field for each state.

Spatial overlays transfer data from fields in a **Source** object set to a **Target** object set using some **method**. The specific rules used to transfer data are taken from the Transfer Rules specified for each column.

**Spatial Overlay Dialog Controls**

**Source**
Choose an object set to be the source of data. Choices will include the selection, all saved selections, and objects in drawing layers when launched from a map. The source set may contain only objects of one type (that is, all areas, all lines or all points).

**Target**
Choose an object set to receive the transferred data. Choices will include the selection, all saved selections, and objects in drawing layers when launched from a map. The target set may contain only objects of one type (that is, all areas, all lines or all points).

**Method**
Choose a transfer method. Only those operations that make sense for the objects in the source and target sets will be presented. If the source or target set do not contain all objects of one type the method box will not be enabled.

**Spatial Overlay Methods**

The following methods will appear in the **Methods** box as appropriate to the types of objects that have been selected in the **Source** and **Target** sets.

- **Areas to contained areas**
  Transfer fields from area A to areas that are completely inside A.

- **Areas to contained lines**
  Transfer fields from area A to lines that are completely inside A.

- **Areas to contained points**
  Transfer fields from area A to points inside A.

- **Areas to containing areas**
  Transfer fields from area A to areas that completely contain A.

- **Areas to intersecting areas**
  Transfer fields from area A to areas that have at least some location within A’s interior (that is not on a boundary).

- **Areas to intersecting lines**
  Transfer fields from area A to lines that have at least some location within A’s interior (that is not on a boundary). Excludes fully contained lines.

- **Areas to neighbor areas**
  Transfer fields from area A to areas that touch A only at A’s boundary (that is, have no common locations that are interior for A).

- **Areas to neighbor lines**
  Transfer fields from area A to lines that touch A only at A’s boundary.

- **Areas to boundary**
  Transfer fields from area A to points that lie on its
Example

Let’s consider a simple example to see how spatial overlays work.
We've created a drawing with two areas and nine points. Select the points and save them as a saved selection called **Points** in the Selections pane. Select the two areas and save them as a selection called **Areas** in the Selections pane.

<table>
<thead>
<tr>
<th>Population</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

The table for this drawing shows that there is one integer field, **Population**. Each of the points has a value of 1 for the population field and is shown with red selection background color in the table. The two areas have values of 0 for the population.

Open the drawing’s table, right click onto the **Population** column and choose **Transfer Rules**.

In the **Transfer Rules** dialog choose **Copy** as the transfer rule for **1 to N** transfers and choose **Sum** as the transfer rule for **N to 1** transfers.

Click onto the drawing window and choose **Drawing - Spatial Overlay**.

In the **Spatial Overlay** dialog choose **Points** as the **Source** and **Areas** as the **Target**. The **Method** will be **Points to containing areas**. Press OK.
The result of the spatial overlays operation is that one area (the triangle) acquires a value of 2 and the other area (the circle) acquires a value of 4. These new values are the sums of the point values within the areas.

Geographic Example

Let's apply the above procedure in a geographic setting. We will sum up the populations of towns to get a total population for each county containing the towns.

We have a drawing called Example that shows counties in the San Francisco Bay area as areas together with points taken from a Census Bureau file of named places with populations.

Opening the Example Table we see that there are ten areas each with a 0 value in the pop1990 field. Numerous towns have populations for each town.

We begin by right clicking on the pop1990 column header in the table and choosing Transfer Rules.
We set the transfer rules for this field to be **Sum** for **N to 1**. Since many points will be combined into each county this is a "many to 1" or **N to 1** transaction. We would like the population values to be summed to get the value for each county.

We don't want any values transferred for the **county** or **Name** fields. We right click onto the **county** column header, choose **Transfer Rules** and choose **None** for the **N to 1** transfer rule. We then right click onto the **Name** column header, choose **Transfer Rules** and once more choose **None** for the **N to 1** transfer rule.

If we were using spatial overlays between different drawing layers in a map we could skip this step; however, since we are transferring values between fields in the same drawing we will use saved selections. Select all points in the drawing and save this selection as a saved selection in the Selections pane called **Places**. Select all points in the drawing and save this selection as a saved selection called **Counties**.

With the focus on the **Examples** drawing window, choose **Drawing - Spatial Overlay**.

In the **Spatial Overlay** dialog choose **Places** for **Source** and **Counties** for **Target**. For the **Method** choose **Points to containing areas** and press **OK**.
In the table we can see that the pop1990 values are filled with the sum of the point values for each county. This is in accordance with the N to 1 transfer rule we specified for this field. Note that there is no change in the county or Name fields for the county areas since we specified None as the transfer rules for these fields.

**Spatial Overlays in Maps**

Spatial overlays may be used in maps when a map contains at least one non-empty drawing. To use spatial overlays between different drawings in a map, the following must be true:

- The same field name and type must be present in both drawings involved.
- Exactly the same set of transfer rules must be used in both drawings for each fields involved.

If the above two conditions are not met for a particular field it will not be transferred at all.

**See Also**

Transfer Rules - Transfer rules specify how fields are aggregated or allotted when new objects are created using transform toolbar operators. Spatial overlays are a different concept than the idea of using transfer rules to aggregate or to allot values when creating objects with transforms. However, it is a related concept in that aggregations or allotments are used to transfer field values between objects.

Transfer Contour Line Height to Points - A simple example using Spatial Overlay.

Shortest Path over Land - A complex example that uses Spatial Overlay.
Drawing - Topology Overlay
See the Topology Overlay topic.
The Transfer Heights command takes heights from a surface and transfers them to objects in drawings (points, lines or areas) that overlay that surface. It is a fast and easy way to get the height of a surface at different points. The Transfer Heights command may be thought of as an analog of the Spatial Overlay command designed for use between surfaces and drawings.

The Drawing - Transfer Heights menu selection is enabled when the optional Surface Tools extension has been installed and the focus is on a drawing window or on a drawing layer in a map window. If you have not activated the Surface Tools extension with a valid Surface Tools serial number you will not be able to use the Drawing - Transfer Heights command.

Launching the Transfer Heights command in the Drawing menu launches the Transfer Heights dialog.

Controls

Source  Choose the surface from which heights will be taken.
Target  Choose the drawing to which heights will be transferred.
[Field] Choose the name of a field in the target drawing's table to which heights will be transferred. If no columns exist in the target table, a choice of [New Column] will create a new column called Height.

Aggregate heights… Specify the method used to assign height values to line and area objects, which extend over many different heights:

Average - Assign the average height found in the line or area.
Maximum - Assign the maximum height found in the line or area.
Minimum - Assign the minimum height found in the line or area.
Sum - Assign the sum of heights found in the line or area.

Example

Suppose we have a drawing of five points showing locations of survey stations in the Montara Mountain region of California. We also have the 1:24K-scale SDTS DEM surface of the region downloaded from the USGS website.
We can show the drawing and the surface together in a map to see where the five points are located relative to the surface.

If we open the drawing’s table we can see that so far there are no fields (except for the default ID field) in the table.

Clicking back on the drawing layer to move the focus there, we can launch the Drawing - Transfer Heights dialog with the above values and then press OK. It doesn't matter what we choose for the Aggregate choice since there are no lines or areas in the drawing.
Instantly, a new field called **Height** will be created in the table and it will be populated with the height of the surface at each point.

**Comments**

Although the above example uses terrain elevation, the **Transfer Heights** command works perfectly well regardless of what data is conveyed by the surface. For example, if we had a surface that showed average temperature the value of average temperature could be transferred as the "height" to objects in the target drawing.

The **Transfer Heights** command is a fast way of noting the height of a surface at points of interest. Simply create a map with the surface and then add a blank drawing layer to the surface. Click with the Insert Point tool to create points where desired and then use **Transfer Heights** to save the height of the surface at each point into the drawing's table.

The **Sum** option for transfer of heights to areas provides a way of aggregating values from surfaces, if those surfaces have been correctly designed for such uses. It locates all the surface's pixels that lie within an overlying area and sums up the value of the height for each pixel.

**Sum** might be used, for example, in a case where a surface represents the concentration of a toxic material deposited downwind of a hazardous material incident (such as an explosion or fire). If the concentration is given in some value per square meter and we know the size of the pixels used to represent the surface, then the **Sum** of values for pixels that occur in a given area will be directly related to the total amount of toxic material deposited within that area. If the area represents, say, a lake or other body of water we can find the total amount of toxic material deposited into that lake.

Note that since clearly, if the same surface is represented with more pixels (that is, each pixel covers a smaller region) then the **Sum** will be greater. However, if we pay attention to the size of pixels in use we can re-scale the result as desired by simply multiplying or dividing the resulting **Sum**.

**See Also**

**Surface Tools**
The **Visible Area** command works with a surface and a drawing to find all parts of the surface that are visible from one or more locations in the drawing marked by points. This function in other GIS systems may be called a *viewshed* function or *visibility zones* function. This function is bi-directional in that it may also be used to find all locations from which the tops of one or more towers are visible from the surface.

The **Drawing - Visible Area** menu selection is enabled when the optional Surface Tools extension has been installed and the focus is on a drawing layer in a map window that also includes a surface layer. If you have not activated the **Surface Tools** extension with a valid **Surface Tools** serial number you will not be able to use the **Drawing - Visible Area** command.

The **Visible Area** command creates a new area object within the target drawing that shows the region of the surface that is visible from the designated points.

Launching the **Visible Area** command in the **Drawing** menu launches the **Visible Area** dialog. The dialog above uses a surface called **Montara Mountain** and a target drawing called **Drawing** (not a very original name, but descriptive).

**Controls**

- **Source**: Choose the surface for which visibility is to be analyzed.
- **Target**: Choose the drawing from which viewing point locations will be taken and within which the visible area will be created.
- **Scope**: [All Objects in drawingname] or the selection or a saved selection, if one exists.
- **Height**: The data field to be used for heights of observing points. Choose [None] if no height field is to be used, in which case the observing points will be taken to lie upon the surface.
  - **Relative Heights**: If unchecked, interpret the height field to mean an absolute altitude. If checked, interpret the height field to mean a relative height above the surface at that location.
  - **Compute area visible from all locations**: If unchecked and multiple points are given the created area will be the region visible from any one of the locations. If checked, the area will represent only that part of the region that is simultaneously visible from all of the locations.

**Using Heights**

If [None] is specified in the **Height** field, Manifold will compute the regions of the surface visible from each location assuming that the observer's eye is located at ground level. That usually is not a realistic choice since a
literally ground level view will easily be obstructed by slight rises that would not impede the view even from the height of an average person's eye.

It is much more frequently the case that we would like to know areas that are visible from a person's eye or from a tower or other structure. In such cases the eye of the observer is elevated to some height above the surface. The **Height** option allows us to specify the name of a column in the drawing's table that provides the heights of each viewing location.

At times the height of the viewing location is known in absolute terms (such as the height of an aircraft observer) and at times the height of the viewing location is known in relative terms (such as the height of a tower above ground level). The **Relative heights** option box allows us to choose which is the case for the height values in our drawing's table. Check this option box to use relative heights, for example, the height of a tower above ground level.

**Example**

This example computes the area visible from five towers in the region of Montara Mountain, California. It uses a drawing of five points and a surface taken from part of the Montara Mountain USGS SDTS DEM data set.

Imported into the project the components appear in the project pane as seen above. The drawing (called **Drawing**) and the surface (called **Montara Mountain**) appear together in a map (called **Map**).

Opening the map we see that the drawing contains five points.
Opening the drawing's table, we see that each point has a **Height** value ranging from 100 to 10. The points have been thematically formatted in the drawing so that the color of each point varies from yellow (the 10 point) to purple (the two 100 points located on ridges).

The drawing represents five towers ranging in height from 10 feet to 100 feet that are located in various parts of the Montara Mountain region.

To find the visible area we click on the drawing tab in the map and launch **Drawing - Visible Area**.

In the **Visible Area** dialog we choose **Montara Mountain** as the **Source**, **Drawing** as the **Target**, [All Objects in Drawing] as the **Scope** and **Height** for the **Height** field. We check the **Relative heights** box because the height for each tower is given as the height above ground level at that position. Press **OK**.
The result is that a new area is added to the drawing. We have taken a moment to format the area in yellow color so that it is easy to see. Any location on the surface within that area is within sight of the top of one of the five towers. Because there are five towers and three of the towers are high towers, much of the surface is visible from at least one of the towers.

Suppose we would like to determine the visible area from only a subset of towers? That’s easy to accomplish.

Let’s suppose we are interested only in the area that is visible from either of the two tallest towers. We delete the new area created in the previous step, and then we select the two tallest towers (the purple dots).
We launch the Visible Areas dialog as before except this time the Scope is set to [Selection in Drawing]. Press OK.

The result is a new area that shows all parts of the surface that are visible from one of the two selected towers. Although it is a smaller area than is visible from all five towers, nonetheless it covers a large part of the surface because the two selected towers are tall towers that are located on high ridges.

So far we have looked at two cases where the visible area is visible from any one of the towers. Suppose we would like to find those regions that are visible from both towers at once?
We could delete the visible area created in the previous step, select the two tall towers and then re-run the Visible Areas dialog. This time we can check the Compute area visible from all locations box.

The result is a new area that shows all parts of the surface visible to both of the two selected towers at the same time. Because line of sight is bi-directional, if we stood at any of the locations marked by the yellow area we would be able to see the tops of both towers.

If we like, we can see which area is visible from all five towers simultaneously.
To do so we launch the **Visible Area** dialog using **[All Objects in Drawing]** for the **Scope** and we check the **Compute area visible from all locations** box.

The result shows that very few locations can be seen from all five towers at the same time. In general, it is only portions of the highest ridges that may be seen from all five towers at once. If we stood at any location within the yellow area we could see all five towers at once.

**Notes**

Only one area object is created by the **Visible Area** command. The area is a branched area object so that even those parts of the area that appear disconnected from other parts of the area all belong to the same area object. Clicking on any one part of the area using touch selection will select the entire area.

Strictly speaking, if we want to consider the visibility of a 100-foot tower we should add some value for the height of the individual who is viewing. One way to do this is to add some average amount, say, 5.5 feet, to the height of each tower.

Line of sight is bi-directional. Every spot that may be seen from the top of a tower is also a spot from which the top of a tower may be seen. The **Visible Area** command not only lets us find all locations that are visible from a given location, it also allows us to find all areas from which the given location is visible. This function is very...
useful in determining whether or not towers or other structures can be seen. For example, if we would like to
determine the areas from which a cellular telephone antenna may be seen we can determine the visible area from
the cellular antenna's location using the height of the antenna. Anyone standing within the visible area will be
able to see the antenna.

This command is very useful for planning any line of sight task. For example, it can be invaluable when planning
the location of monitoring instruments (such as seismic sensors on volcanoes) that connect to central stations via
microwave or laser communications links, the location of observation posts for borders or secure installations or
even the location of defensive positions in military installations.

Computing visible areas can be a highly computationally intensive task that can take a long time to accomplish
with large surfaces or with many viewing points.

See Also

Surface Tools

Labels Menu

Labels - Text

The Labels - Text menu appears for bound labels components that are created from fields in a parent drawing.
The dialog allows us to change the text strings and fields used to create labels in the component.

The upper pane of the dialog contains fields available in the drawing's table, including intrinsic fields. Double-
click on any of the fields in the upper pane to add it to the text in the lower pane.

See the Creating Labels from Fields topic for information on creating bound labels.
Labels - Synchronized

A labels component is synchronized when it has been created from a drawing with the Automatically label new drawing objects checked on. In that case, whenever a new object is added to the parent drawing a new label will be created for that drawing. Whenever an object is deleted from the parent drawing the label associated with that object will also be deleted. Labels cannot be manually added to or deleted from a synchronized labels component, because the labels in that labels component are automatically created or deleted based upon objects in the parent drawing.

The Labels - Synchronized option toggles labels synchronization on (checkmark) or off (no checkmark). When synchronization is off (no checkmark), then although the labels component is still bound to the parent drawing, labels will not be automatically created or deleted as objects in the parent drawing are created or deleted. This allows us to delete some of the labels, to copy and paste just some of the labels into a new bound labels component and to use the Insert tools to create labels for only some of the objects in the parent drawing. See the Synchronized Labels topic for details and examples.

See Also

Editing Labels
Labels
Synchronized Labels
Labels - Unlink

The **Labels - Unlink** command is enabled for labels components that are bound to a parent drawing. A **bound** labels component creates labels based upon objects in a parent drawing and it appears in the project pane hierarchy indented underneath the parent drawing.

If we **unlink** such a labels component we convert the labels within it to ordinary, unbound labels that no longer have any relationship to objects in the parent drawing or to any data attributes for objects in the parent drawing. The current state of whatever text is in those labels based upon fields in the parent drawing (if such are used) will be copied to static text within the labels. The now unbound labels component will move in the project pane hierarchy to the same level as any independent component and will no longer appear indented underneath the parent drawing.

Unlinking a labels component is a "one way" operation. Once unlinked, the labels component cannot thereafter be re-linked to the former parent.

See Also

Creating Labels from Fields
Editing Labels
Labels
Synchronized Labels

Tools Menu

**Tools - Add-In Manager**

The **Add-In Manager** dialog allows specifying which Manifold System add-ins will be operational for this particular user. In addition, one add-in can be designated to guide the startup configuration of Manifold system.

Add-ins are installed for all users on the computer system. Newly installed add-ins will be enabled for all users. Individual users can open the **Add-In Manager** dialog to specify which add-ins should be loaded for their user name and which add-in will specify toolbar configuration.

The **Add-In Manager** dialog shows a list of all add-ins available in the system with checkboxes next to each. Check the box of an add-in to load it for use when Manifold launches. Uncheck the box to instruct Manifold not to load or use an add-in for this user.

Select All  - Check all add-ins.

Select None  - Uncheck all add-ins.

Select Inverse  - Uncheck all checked add-ins and check all unchecked add-ins. A fast way to use all but one add-in: click **Select None**, check the one add-in not desired and then click **Select Inverse**.

Add-In  This pane lists add-ins available in the system. Check an add-in to make it operational within the system.

Config  Only one add-in can control the startup configuration of toolbars. Double-click into the **Config** column to change the config status of an add-in to **Yes**.

Status Bar  The status bar will show any copyright or comments text embedded into the add-in that is highlighted in the **Add-In** pane. Click on an add-in to highlight it.

After changing settings Manifold System must be exited and then restarted for the new settings to take effect.
See the Add-Ins topic for information on creating add-ins.
Tools - Administrator Console

This topic describes a feature that is provided only within Manifold Database Administrator Edition and is not available in other Manifold System editions. If you have purchased and installed Database Administrator Edition the features described in this topic will be available to you. If you have installed some other Manifold System edition you will not be able to use these features.

The Tools - Administrator Console dialog provides a user interface for DBMS administrators to configure and manage Manifold geospatial data storage within enterprise DBMS data stores. Once database objects are configured with Administrator Console, the enhancements and ease of use features will be available to other users using other Manifold System editions.

Controls

Data Source
The name of the last used data source. Manifold will not actually connect to the last used data source until the Refresh button is pressed. Press the browse button to launch the Data Sources dialog to choose a data source.

(contents pane)
Displays the contents of the data source.

Refresh - Update the contents pane.

Properties - Launch the Properties dialog for the highlighted database object. Allows specification of the friendly name, precision, version column, formatting and server version incrementing options. Also used to configure tables for treatment as drawings with specification of geometry columns and geometry data type.

Triggers - Launch the Triggers dialog for the highlighted database object. View and edit triggers bound to the database object (Oracle only). Shows the trigger, type of trigger and event for each trigger as well as the procedure text.

Run - Press to execute the command line text in the command pane.
Global Filter - Launches a dialog enabling filtering the names of database objects using regular expressions. Changes to the global filter are saved between different sessions of Manifold. By default, the global filter includes several masks suppressing display of system tables commonly found in Oracle databases. See the Tools - Database Console topic for use of the Global Filter button.

Columns - Toggle display of desired columns on or off in the contents pane. Manifold will remember the last used columns between sessions. Data is fetched for columns from the data source on demand.

Columns include:

- **Component** - Show the "friendly" name, that is, the Manifold component name.
- **Format** - Enable storage of formatting for drawings. Double-click into the Format cell of a database object to enable or disable formatting.
- **Precision** - Show location precision used in drawings.
- **Projection** - Projection used by this database object. Double-click into the Projection cell of a database object to launch the database object Projection dialog.
- **Spatial Index** - Reports if a generic spatial index is present for this object. Double-click into the Spatial Index cell of a database object to launch a dialog that enables creating and editing the spatial index. The bounding box of the area covered by the index is editable. A table must have a primary key to allow creation of a spatial index.
- **Statistics** - Useful information, such as the number of objects in drawings or the number of records in tables.

**Caution:** Do not show the Statistics column when database tables may contain very many records. It can take a very long while for some DBMS products to report how many records are in a given table.

- **Triggers** - The number of database triggers, if any, for this object. Double-click into the Triggers cell of a database object to launch the database object Triggers dialog.

Filter

Filter by character sequence - Show only those database objects in the contents pane the name of which contain the given sequence of characters. This is particularly helpful when working with data sources with lots of objects, such as SQL Server or Oracle data sources. See the Tools - Database Console topic for use of the Filter box.

(command pane) A text pane in which command lines, such as SQL, can be entered to be executed by the data source.

(status bar) Information about the highlighted object will be provided at the bottom of the dialog.

### Spatial Index Dialog Controls

The spatial index dialog appears when double-clicking into the Spatial Index cell for a database object that Manifold has been instructed to treat as a drawing.

**Object** - The database object to index. This will be a table name followed by a dot followed by the geometry column to be indexed. A typical name in SQL Server Express, for example, might be `dbo.Mexico Table.GeomI` if we exported a table using the `Geom (I)` intrinsic field to provide a geometry column.
Use spatial index  
Check to create a spatial index.

X and Y boxes  
Four boxes for the X and Y extents to be covered by the spatial index, initially blank until the Suggest button is pressed. The outer two boxes (not editable) show the extents of the data set. The inner two boxes show the extent of the spatial index to be constructed. A spatial index must include all objects in the drawing. Attempting to create a spatial index with bounds that do not include all coordinates of all drawing objects will fail.

Level  
Degree of detail in the spatial index, providing a balance between granularity and performance. Higher values produce more granular indices, which require more space but are potentially more efficient. For best efficiency, choose an index level proportional to the number of objects expected to be in the drawing. The value suggested by the Suggest button is a reasonable default.

Suggest  
Press this button to command Manifold to analyze the data set to determine the extents of the data and a recommended

For details on using the Administrator Console, see the Using Administrator Console topic.

Tech Tip

Keep in mind that the choice of connection technology used in the Data Source will influence how Manifold interacts with the DBMS. For example, ADO .NET connections typically are read-only and will not allow either Administrator Console or Database Console to make changes, such as adding a spatial index, to the database.

See Also

Database Administrator Edition
Database Object Projection
Database Object Properties
Database Object Triggers
Spatial DBMS Facilities
Tools - Database Console
Using Administrator Console
Tools - Database Console

The Database Console allows browsing ADO.NET, OLE DB, ODBC or file-based (like .mdb or .xls files) data sources to see what tables are available and what fields and type of fields they contain. Connections may also be made to Oracle databases using the Oracle Call Interface (OCI) for a native Oracle connection, or to spatial DB2 or PostgreSQL databases using a native DB2 or PostgreSQL connection. Toolbar buttons allow importing or linking tables from data sources.

The Database Console is resizable, so we can increase the size of the dialog if we have many objects to display or a lengthy query to write.

The database console understands a variety of spatial DBMS storage methods:

- **Native spatial DBMS** - When using the DBMS's native connection technology to connect to a DBMS providing "built in" spatial capabilities such as Oracle Spatial, IBM DB2 with spatial extender, SQL Server 2008 or PostgreSQL with spatial extensions, database console will automatically utilize the built in spatial features of that DBMS. Images stored as Oracle GeoRasters will also be automatically identified as images.

- **Manifold spatial DBMS** - When connecting to a DBMS used to store spatial information using non-native geometry types supported by Manifold, database console will automatically use Manifold spatial indices and interpret tables containing geometry as drawings. Images and surfaces stored using Manifold facilities will also be automatically identified as images and surfaces. Spatial indices in SQL Server managed by the Manifold Spatial Extender for SQL Server will automatically be recognized and used.

- **ESRI SDE or Personal Geodatabase storages** - If SDE or Personal geodatabase storages are present in the database, database console will recognize those and will configure itself to work with them.

The above are referred to as spatial databases for the purposes of this topic. A spatial database is one of the above three classes of database and is distinguished by having tables that contain geometry for drawings using generic geometry types and generic spatial indices. Although any Manifold edition can read generic spatial DBMS storage enabled with Manifold facilities, only Manifold System Enterprise Edition or greater edition can create such storage.

To work with spatial DBMS using native spatial DBMS capabilities (such as drawings and images stored within Oracle Spatial or Locator facilities within standard Oracle databases), we must have Manifold System Enterprise Edition or greater installed. If we do not have at least Enterprise Edition installed, we will not be able to connect to and use native spatial DBMS capabilities within spatial DBMS products such as IBM DB2 with Spatial Extender, Microsoft SQL Server 2008 (Katmai), Oracle Spatial or PostgreSQL with spatial extensions.

The database console dialog also allows interactive execution from within Manifold of command language statements to manipulate the external database. This allows one to completely control a large data source (such as an Oracle or SQL Server database) from within a Manifold dialog.

**Note:** the example screenshots in this topic use a connection to the Northwind Traders sample Access database that has been set up in the Data Sources dialog. See the examples in the Data Sources Dialog topic.
The database console dialog consists of two panes. The upper, contents pane allows browsing of all tables, fields and queries in the database, including tables in addition to the table that is opened. The bottom pane provides a workspace for writing statements in the command language for that data source (normally SQL/DDL). Database objects in the upper pane are sorted by type (tables are shown together, queries are shown together, etc.) and then by name. Table columns are sorted by name.

Highlighting an item by clicking on it will show what it is. For example, the highlighted table in the illustration above is identified as a TABLE in the small status row at the bottom of the dialog. If the provider supports comments or descriptions for fields, these will appear in tool tips when the mouse hovers over a field. In the illustration above (showing the Nwind.mdb sample database) the description entered into the table design for the Customer ID field is shown as a tool tip when the mouse cursor hovers over that field.

Field types will be shown for any fields highlighted in the contents pane. Field types will be reported using Manifold equivalents so we can see what the column type will be when the table is imported or linked into a Manifold project.

When connecting to a spatial database any tables that contain one or more geometry columns will be shown as drawings in the contents pane. Such drawings may be imported or linked into the project. For example, when connecting to an Oracle database using OCI, any tables that contain one or more SDO_GEOMETRY columns will be shown as drawings in the contents pane. To take another example, when connecting to any database using OLE DB which has a spatial index created by Manifold, any tables that contain one or more geometry columns of
any supported Manifold geometry type (such as OGC Geometry(WKB) or Manifold Geometry) for which a spatial index has been declared will be shown as drawings in the contents pane.

Likewise, images stored either using native DBMS technology or generic Manifold image storage technology will appear with image icons and may be imported or linked into the project. For example, images stored in Oracle databases using GeoRaster technology will appear with image icons. The status readout for an image stored in an Oracle database will show the dimensions of the image, the number of channels and the data type used for image pixels. Surfaces will also appear as images: surfaces may be imported by importing the Height channel of the image.

**Data Source**
The name of the last used data source. Manifold will not actually connect to the last used data source until the Refresh button is pressed. Press the browse button to launch the Data Sources dialog to choose a data source.

**Refresh**
- Update the contents pane.

**Import**
- Imports a copy of the highlighted table into the project. The table will be stored within the Manifold project with no connection to the originating data source. Also used to import drawings from geometry tables in databases or to import images or surfaces from GeoRaster storage in Oracle databases.

**Link**
- Links the highlighted table into the project. The table remains stored in the data source. Also used to link drawings from geometry tables in databases or to link images from GeoRaster storage in Oracle databases.

**Component View**
- Enabled if the Administrator Console has been used to add friendly, component names to this database for database objects. Press in (the default if friendly names are available) to show the database using component names as in a Manifold project. Database objects without friendly names will be hidden.

**Global Filter**
- Launches a dialog enabling filtering the names of database objects using regular expressions. Changes to the global filter are saved between different sessions of Manifold. By default, the global filter includes several masks suppressing display of system tables commonly found in Oracle databases.

**Filter**
- Filter by character sequence - Show only those database objects in the contents pane the name of which contain the given sequence of characters. This is particularly helpful when working with data sources with lots of objects, such as SQL Server or Oracle data sources.

**Run**
- Press to execute the command line text in the command pane.

**Import Query**
- Fetches a copy of the results of the query into the project as a table. The table will be stored within the Manifold project with no connection to the originating data source.

**Link Query**
- Links the results of the query into the project as a table. The data remains stored in the data source.
Note: By default, system tables and views do not appear in the data source contents pane for Access .mdb databases. To cause them to appear, open the Tools - Options - Import and Export dialog and check the option box for Show system tables and views in external data sources.

When connecting via ADO .NET if the database provider supports schemas (introduced in ADO .NET 2.0) then the database console will list tables and columns returned by the database provider.

Friendly Names

Enterprise-class databases used for storing drawings and other data from Manifold will often employ fearsome naming conventions for database objects that, to put it mildly, ordinary users will find threatening. Friendly names, also known as component names, make it possible to use simple, non-threatening, names within the database just like in our Manifold project.

To add friendly names to a database, an administrator must use the Administrator Console to add them. Normally, this is done at the same time a drawing or other Manifold component is first uploaded into a database. If the Administrator Console has not been used to specify friendly names for use in a database, then friendly names will not be available.

When Database Console connects to a data source that has had at least one friendly name assigned with the Administrator Console, Database Console will automatically launch with component view turned on, that is, to display friendly names by default. If for some reason component view has been turned off, we can turn it back on by using the Component View toolbar button. If the data source does not contain a metadata table set up with Administrator Console, the component view button will be disabled.

When using component view, the Database Console dialog will only show components that have been assigned friendly names. The Filter box will work with friendly names as well.

For example, if a drawing is exported into an Oracle Spatial database it might be known within the database as SYSTEM.US_MAINDRAWING.GEOMETRY. In normal usage, of course, everyone would prefer to call that drawing by a simpler name such as US Main.

For example, without friendly names enabled we have to wade through many database objects that are not of interest even if global filtering is set.

With friendly names, when Component View is turned on, we see only those database objects usable as Manifold components as we might in the project view, without any unnecessary clutter.

Note: The Administrator Console is available only in Database Administrator Edition licenses of Manifold System. The ability to enable use of friendly names by the entire organization is one of the many reasons why it makes sense for organizations that use Manifold System to acquire at least one Database Administrator Edition license. Once a Manifold Database Administrator Edition license has been used to create friendly names in a database, then clients using other Manifold licenses, such as Enterprise Edition licenses, can use those friendly names. Therefore, organizations with many Manifold licenses will normally have a few Database
Administrator Edition licenses for their DBMS or IT administrators to use with most of their Manifold licenses being Enterprise Edition licenses.

Resolving Identical Names

If a database object has been assigned the same friendly name then regardless of the Component View setting the Database Console will display the object

Import / Link Options Dialog

Clicking the Import or the Link button will import the highlighted component. If the Administrator Console has been used by a DBMS administrator to pre-configure the import and link options for the component, it will import or link immediately from the database.

Any Manifold edition can connect to a spatial DBMS managed by Manifold using either generic spatial indices or SQL Server running the Manifold Spatial Extender for SQL Server. Enterprise Edition or higher edition is required to export components into such databases, but once the components are uploaded into the database and a spatial index created (for drawings), then any Manifold System edition can import or link that data.

Enterprise Edition is always required to connect to a native spatial DBMS such as DB2 with IBM Spatial Extender, Oracle Locator or Oracle Spatial, PostgreSQL or SQL Server 2008 spatial.

If Administrator Console has not been used to pre-configure import and link options, clicking the Import or the Link button will launch the Import / Link Options dialog to allow specification of options during the import or link process. Most options require spatial DBMS capability, either using a native spatial DBMS or Manifold-managed spatial indices.

<table>
<thead>
<tr>
<th>Name</th>
<th>Name to use for the component that is imported or created.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use all objects</td>
<td>Spatial DBMS only. Import or Link all objects.</td>
</tr>
<tr>
<td>Use all objects in the following area</td>
<td>Spatial DBMS only. Import or link only a subset of objects found within the given X (longitude) and Y (latitude) extents. The read-only X and Y coordinate boxes that surround the editable X and Y coordinate boxes report the minimum and maximum X and Y extents of the data as reported by the data source.</td>
</tr>
<tr>
<td>Require objects to be completely within area</td>
<td>Spatial DBMS only. Enabled if a subset area of interest has been specified. Not checked by default: will import or link any object that has at least one coordinate within the given area of interest. If checked, will import or link only those objects that have all coordinates entirely within the given area of interest.</td>
</tr>
<tr>
<td>Version</td>
<td>Enterprise Edition only. Column to use as the version column for tracking and resolving concurrent multi-user editing conflicts.</td>
</tr>
<tr>
<td>Rely on server to increment version automatically</td>
<td>Spatial DBMS and Enterprise Edition only. Enabled when a Version column is specified. If checked (the default) causes the database (such as Oracle) to increment the version value whenever an object is edited by any user.</td>
</tr>
</tbody>
</table>

Note that some options require Enterprise Edition and others require both use of a spatial DBMS and also Enterprise Edition.

For example, importing or linking a drawing via Database Console will automatically retrieve and use the value for location precision if available. Drawings uploaded as tables into a spatial DBMS from Manifold will store location precision.
Some data sources contain very large numbers of objects such as tables and queries. It is very convenient in such cases to restrict the items shown in the contents pane to only those of interest. We can do so by entering characters matching the desired names into the text box in the toolbar.

For example, if we look at the sample Northwind Traders database without any characters in the text box we see listed in the contents pane everything in the database.

If we enter the character c into the text box we see listed only those objects that have a c in their name. For example, the Suppliers table no longer appears because there is no c in "Suppliers."
If we enter the character sequence `ca` into the text box we see only those objects that have a `c` followed by an `a` in their names.

**Global Filter**

The Global Filter provides a default way of filtering out system level and other DBMS objects we don’t need to see.

Pressing the Global Filter button opens the Filter dialog. The lower pane of the dialog contains a list of masks, which are simply regular expressions using standard regular expression syntax. For example, the `*` asterisk character matches zero or more characters of any type.

Object names that match any of the regular expression masks within the Filter dialog will not be displayed. If we have a mask like `OLAP.*` then any database object that has the letters `OLAP` followed by zero or more characters will match. Such a regular expression matches names like `OLAP1` or `OLAP` or `OLAP.234`. So DBMS names like `OLAP.INDEX` or `OLAPSYS.XML_LOAD_LOG` or similar will not be displayed. See the Regular Expressions topic for details on regular expressions.

To add a new mask, enter the regular expression in the uppermost box in the Filter dialog and press the Add button. To delete a filter, highlight it in the list of filters and press the Delete button. The Add button is enabled whenever text is entered into the upper box, and both the Add and Delete buttons are enabled if any existing mask is highlighted.

**Executing Commands**

The lower pane of the database console allows direct control of databases using the command language (normally SQL or associated languages such as DDL) supported by the provider. For example, when connecting to a SQL Server Express Edition server or to an Oracle Express Edition server, database console is a handy way of creating new tables.

The lower pane of the database console may also be used to execute queries and to import or link the resulting tables into the project. This is an alternate, interactive way of creating tables from queries in addition to creating a Query component.

After creating a query in the lower pane we press Run to execute the query.

**Editing Shortcuts**

The command pane in the lower part of the dialog provides an edit pane into which command statements may be written. For convenience, double clicking a database object in the upper pane will add it to the text in the lower pane at the current cursor position.
For example, suppose we open the database console with the *nwind* sample database and begin writing a query that will use the *Customers* table. We would like to select company names. To do so we double click the *Company Name* field object in the upper pane.

It appears in the lower pane as part of the query. When automatically adding names to a query in this way, Manifold will insert the name of the object into the query using the correct quote characters for the type of database connection being used.
We can continue writing the query by entering `FROM` into the lower pane and then double clicking the `Customers` table object. in the upper pane.

```
SELECT 'Company Name' FROM `Customers`
```

This adds "Customers" to the query text in the lower pane. The ability to add text to the query in the lower pane by double clicking objects in the upper pane is a great way to avoid entering long names manually and also helps avoid typographic errors.

Manifold is reasonably adaptive when loading database object names into the lower pane. For example, double-clicking an image in a spatial DBMS will insert the name of the database table containing the image data.
Using Run makes sense only if the SQL command is an action query used to alter table structures, etc., since there is no way for a SELECT query to report its results within the database console. The SELECT query in the above sequence of editing shortcuts, for example, would have no effect when using Run.

To run a SELECT query like the one above and collect the results in a useful way, press either the Import Query button or the Link Query button so that the results of the query appear as a new table in the project.

For example, if we press the Link Query button with the command text above we will create a linked table in the Manifold project that lists the city and country for each record in the Customers table, as seen below.

After running an action query, press Refresh to update the upper pane in the dialog (necessary if the SQL statement adds or deletes tables or otherwise alters database structure). If a parameterized command is issued in the edit pane the dialog will prompt for values of required input parameters.

The language available within the database console will be whatever language is supported by the underlying OLE DB driver. The execution capability of the database console is therefore driver-specific.

When running SQL within Manifold queries, one is using the Manifold SQL engine. When executing SQL within the database console, one is using whatever SQL is the native SQL of the external database system. One should be aware that SQL implementations in various database systems may contain bugs. If an SQL bug occurs within the database console, the bug should be tracked down with the vendor of the external database system being used.

Examples

Connecting to the Nwind.mdb sample database we could create a new table, alter it and then delete it using the following sequence of commands in the lower pane of the database console, pressing Execute after each command.

```
CREATE TABLE People (Name TEXT, Age NUMBER)

ALTER TABLE People ADD COLUMN Weight NUMBER

DROP TABLE People
```

We can work with other databases through the database console, for example, Microsoft's SQL Server 2000. For examples showing configuration of Microsoft's SQL Server 2000 desktop engine see the SQL Server 2000 desktop engine topic.
Open Data Source Command

Right clicking a component that is linked from an external data source in the Project pane and then choosing **Open Data Source** will open the Database Console and connect to the component data source.

Opening a component linked from an external data source in its own window, and then choosing the **Open Data Source** command in the component menu (for example, choosing **Drawing - Open Data Source** when a linked drawing window is open) does the same thing.

Losing a Connection

Losing a connection to a data source (such as might happen if a network connection to a remote database is interrupted) will disable the object tree in the upper pane as well as the query text lower pane.

Troubleshooting

Sophisticated DBMS installations can very easily be configured with all sorts of security and access permissions that can be difficult to understand. Therefore, if attempting to connect to a database and the connection is not successful, examine very carefully the administrative settings used for the database and connections thereto.

As a confidence-builder in complex situations, it may be helpful to set up a database and connect to it using Manifold to prove to yourself that all is correct with the Manifold installation and with your understanding of Manifold.

For example, install SQL Server Express and connect to it using a data source as given in the Data Sources Dialog topic example.

Note that when specifying a connection in the Data Sources dialog Manifold uses Microsoft configuration dialogs. Manifold uses standard Microsoft technologies, so the key to debugging any connection problems is to understand those standard technologies.

Tech Tip: Connection Methods

Keep in mind that the choice of connection technology will influence how Manifold interacts with the DBMS. For example, ADO .NET connections typically are read-only and will not allow either Administrator Console or Database Console to make changes, such as adding a spatial index, to the database. In another example, to use a spatial DBMS’s native geometry type we must connect to that spatial DBMS using the native connection technology (such as using OCI to connect to Oracle). See the Spatial DBMS Facilities topic for additional information.

Tech Tip: Deletions in Database Console

It’s easy to delete unwanted database tables by writing a drop table query in the query pane at the bottom of the Database Console.

```
drop table SYSTEM.IS_MAIN
```

Simply enter `drop table` and then double-click on the table (drawings and other images are stored as tables in Oracle) in the upper pane to add that name to the query without need to manually enter it. Press the ![run button](run.png) to launch the query and the table will be deleted. Note that the upper list of objects in the database will not be refreshed to show the deletion until we press the Refresh button.
**Tech Tip: Deleting Spatial Components in Database Console**

Using Manifold to store drawings, images or surfaces within a spatial DBMS opens the door to great flexibility, but that flexibility also makes it more complex to delete such components when stored into a spatial DBMS. Storing a component into a DBMS involves storing data in tables within that DBMS and forming metadata relationships with the MFD_META table used by Manifold to keep track of such components. It also makes it possible for other applications or other users to reference the data tables and otherwise form connections to the components in ways that can potentially be very complex.

For that reason, there is no automatic way to delete a component uploaded into a DBMS by just highlighting it and pressing a delete button. Instead, we must follow a simple procedure to delete a drawing, image or surface uploaded into a DBMS:

1. In Database Console, turn off Component View (friendly names) so we can see the full DBMS name of the item to be deleted. Note the name of the object, for example, `dbo.Mexico`.

2. Delete the tables for the component using a DROP TABLE query as given above. This will include the component as well as any spatial index created for it. Hit the Refresh button to see what has been deleted. If an error message such as "Cannot drop the table 'dbo.Mexico', because it does not exist or you do not have permission." pops open, that is a sign that references to that table should be removed from the MFD_META table.

3. Delete all references to that object name within the OBJ field of the MFD_META table. There are two ways to do this:

   a. In Database Console run the following query: `DELETE FROM "MFD_META" WHERE "OBJ"='dbo.Mexico'`

   b. Alternatively: In the Database Console's Filter button, temporarily uncheck the Hide objects matching the following masks box so that all tables are visible, even those we don't normally need to see. Use the Database Console to link the MFD_META table into the Manifold project. This table will appear as a database object with a name like (to use SQL Server as an example) `dbo.MFD_META`. Close the Database Console and use the Query Toolbar or mouse selection to select all records in the table where the OBJ field is equal to the value `dbo.Mexico` and delete those records. After finishing work with this linked table, close it and delete it from the project [deleting it delete the link, not the table itself within the DBMS].

Experienced Manifold DBMS administrators will usually use the second way of eliminating OBJ field references to a deleted table. It's easy to open the MFD_META table in a Manifold project, click on the OBJ column header to sort by the values it contains, and then click once and shift click twice to select all desired values and then press the delete button. This is a quick way of getting rid of similarly-named references to a drawing, the drawing's spatial index and so on. This is quicker and less prone to typographic errors than repeatedly entering a sequence of queries to be executed to delete unwanted records from the MFD_META table.

See Also

- Tools - Administrator Console
- Importing and Linking Tables
- Oracle Spatial Facilities
- Project Pane - Open Data Source
- Oracle Express Edition
- Spatial DBMS
- Spatial DBMS Facilities
- SQL Server Express Edition
- The Data Source Dialog
- Queries
Tools - Make Image

The Tools - Make Image command (keyboard shortcut F6 key) is used to create images from components. It operates slightly differently depending on whether the component is a print Layout or other component. Layouts are different than ordinary component windows because a layout is directly associated with a specific paper size as printed on a specific printer. Within layouts the key question is what resolution (DPI) should be used for the image since the physical extent of the image is fixed by the page size and the composition in the layout.

On the other hand component windows other than layouts do not have any fixed extent nor are they tied to a specific paper size. Drawings, for example, have limitless extent. When an image is to be created based on a component window there are several different ways in which the extent, scale and resolution of the image can be defined.

When Make Image is launched from a component other than a layout, it allows a choice in the Paint dialog of different ways to output the image based on what is in the display window. When Make Image is launched from a layout, the resolution of the image is specified in DPI.

Controls when used from Layouts

Name
Choose a name to use for the new image. The default name suggested will be constructed from the component name that is displayed in the active window.

Description
Any text entered here will appear in the image's Description property in the View - Properties dialog. Use for whatever user-supplied notes may be desired.

Resolution
Choose the desired image resolution in DPI.

Page
Which page of a multipage layout to capture as an image.

(Status bar)
The status bar region at the bottom of the dialog reports the vertical and horizontal size in pixels and the overall size in megabytes of the image that will be created.

Use Make Image from a layout when an image must be created at a specific DPI. For example, suppose we would like to contribute an illustration to a publication and we have been asked by the publication's art department to provide an image at 600 DPI. Suppose the illustration desired is an image of a map component we have created. We can create a layout using the map and then use Make Image from the layout with a Resolution of 600 DPI. We then open the image and export it to whatever image format file (such as TIFF or JPEG) that is required by the publication.

Controls when used from other Components

Name
Choose a name to use for the new image. The default name suggested will be constructed from the component name that is displayed in the active window.

Description
Any text entered here will appear in the image's Description property in the View - Properties dialog. Use for whatever user-supplied notes may be desired.

Paint
Choose the image output from centered view at current scale, current window, entire component and entire component at current scale.

Size
Horizontal x vertical size of the image to be created in pixels. Enabled when the Paint selection is centered view at current scale or entire component.

(Status bar)
The status bar region at the bottom of the dialog reports the vertical and horizontal size in pixels and the overall size in megabytes of the image that will be created.

Paint Options
**centered view at current scale**

Creates an image at the current scale (zoom level) that shows as much of the image as is possible at the given width and height in pixels. Increasing the size (width and height) will show more of the component in the image.

**current window**

Makes a snapshot of the current window at screen resolution (72 or 96 DPI, typically). The image will reproduce what is seen in the window. This option is very fast.

**entire component**

Creates image at the given width and height in pixels that shows the entire component. Increasing the size of the image by increasing the width and height will still show the entire component, but with greater resolution.

**entire component at current scale**

Creates an image of sufficient height and width to show the component at the current scale (zoom level).

Components like drawings have limitless extent. The "entire component" in this context means the size of the minimum enclosing rectangle that encloses all objects in the drawing. When **entire component** is chosen as the **Paint** option, Manifold will fit the rendering of the drawing to the maximum possible size given the pixel size of the image in width and height.

Use **Make Image** from components other than layouts when an image must be created to a particular scale or of a particular width and height in pixels. For example, suppose we would like to create an image from a highly detailed map that we will use as a wallpaper background image on our Windows desktop. Let's say we have a cool new flat panel monitor with 1900 by 1200 resolution. We can open the map, zoom to about the resolution we would like and then use **Make Image** with the **centered view at current scale** option with width and height set to 1900 by 1200 pixels. We then open the image and export it to the file format we want to use.

Note that to achieve a more detailed, higher resolution view we must use **entire component** with a larger number of pixels. Increasing the number of pixels used with the **centered view at current scale** option does not result in more detail, because the level of resolution is set by whatever scale is being viewed (the "current scale").

**See Also**

See the print Layout topic for layout creation. Layouts are used to print items at desired scales.
Tools - Server Console

The Server Console dialog appears in Manifold System Enterprise Edition. It is used to fetch components from Enterprise servers. Given the name of an Enterprise server data source the Server Console displays all of the components available in that server within a tree diagram. Components may be organized within folders to provide better organization for Enterprise servers that contain many components. Click on a component to highlight it and to enable the toolbar.

Server Choose the name of an ODBC data source to connect to an Enterprise server or specify the connection string. The Server box will show the last used name or connection string. Any PASSWORD and PWD parameters used in a connection string will be masked in the last used name or connection string to protect security.

Refresh - Update the components pane. After choosing a new server one must press the Refresh button.

New Folder - Create a new folder in the tree diagram at the current level. Folders may be nested with folders. To move a component into a folder, drag and drop it into that folder.

Import Component - Fetches a copy of the component into the project as a local component.

Link Component - Links (gets) a component from the Enterprise server into the project as a read-only component. The component remains stored in the Enterprise server.

Delete - Deletes a component from the Enterprise server.

Undo Check Out - Enabled when a component is highlighted that is checked out by the same user in a different project. Such components are shown with a red link icon. Undo Check Out will force such components to be shown as checked in on the Enterprise server. The next time the project in which they are checked out is opened or refreshed the system will automatically perform an Undo Check Out on the components.

(Tree diagram) Displays all components available within this Enterprise server. Components already linked within the project will show a dark gray link icon next to the component icon. Components that are checked out in another project belonging to the current user will be shown in a red link icon.

(Status bar) The status bar region at the bottom of the dialog will report the status of any highlighted component. Components already linked within the current project will be reported as Linked. Components within the Enterprise server but not in use in the current project will be reported as Remote. Components that are checked out in another project belonging to the current user will be reported as Checked out in another project.

Simplified Syntax for SQL Server Connections

Enterprise dialogs allow a simplified connection string syntax for SQL Server databases as an option to full ODBC syntax. \"\systemname\" will connect to the default database on a given system. \"\systemname:database\" connects to the specified database on the given system. The connection established with a simplified connection string that uses Windows integrated security (suitable for users working in Windows Server 2003, Windows XP and Windows 2000). Using simplified connection strings is a fast way to connect to SQL Server or to the SQL Server Database Engine using default options without dealing with the ODBC dialogs.

For example, if we have installed SQL Server on a system called PROJECTS and we have created a database called hydrography to use as our Enterprise server database, we could connect in the Server Console by
entering `projects:hydrography` into the Server box of the Server Console and then pressing the Refresh button.

**Tech Tip:** The Server Console is resizable, so we can increase the size of the dialog if we have many components to display.

**See Also**

**Enterprise Edition**
Tools - Batch Export

This topic describes a feature that is provided only within Manifold Database Administrator Edition and is not available in other Manifold System editions. If you have purchased and installed Database Administrator Edition, the features described in this topic will be available to you. If you have installed some other Manifold System edition you will not be able to use these features.

The Tools - Batch Export command works with Oracle databases only. It allows users of Database Administrator Edition to export many drawings, images, and surfaces at once into an Oracle database. The Oracle server must support SDO_GEOMETRY for upload of drawings, and it must support GeoRasters for upload of images and surfaces.

The Batch Export dialog combines the functions of File - Export dialogs for drawings, images, and surfaces into a single dialog, using similar controls. The Batch Export dialog is resizeable.

Control Panel

The Batch Export dialog contains the following controls:

### Server

The name of the Oracle server. oracledb is the default if the full enterprise Oracle product was installed using the directions in the Installing Oracle topic. In the case of Oracle Express installations the server name is xe by default. See the directions for connection in the Installing Oracle topic.

### User Name

An authorized user name to login onto the Oracle server. system is the default system administrator name if the server was installed using the directions in the Installing Oracle topic. Experienced Oracle IT managers will provide users with their own login names.
**Password**  
Password to use to login to the Oracle server.

**Connect**  
Press this button to connect to the Oracle server and to enable dialog controls. Some other controls, such as the OK button, will not be enabled until a connection has been established.

**Select All** - Check all components.

**Select None** - Uncheck all components. No components will be exported. Tech tip: Use this to uncheck all components before checking those desired.

**Select Inverse** - Uncheck all previously checked components and check all previously unchecked components. This is a fast way to check only a few components out of many: uncheck those desired and then push **Select Inverse**.

**Component pane**  
A list of components in the project available for export. Checking the box of a component will export it. Some controls in the dialog will not be enabled unless a component of that type has been checked. Most column values are editable. To edit a specific column value, double-click it. Columns in the pane include:

- **Component** - The name of the component in the Manifold project.
- **Export As** - Name to use for this component in the database. Friendly names may be specified in this column.
- **Precision** - Location precision parameter to be used for drawings. Expressed in whatever native drawing units are used by that drawing.
- **Projection** - Coordinate system to be used within Oracle for this component. Manifold will use the same projection used by the drawing if it is supported by the server (nearly always possible) or whichever is the most similar projection available on the server. Double-click to launch the **Projection** dialog that allows choosing an Oracle projection. Using a projection on the server different than that used by the drawing will require re-projecting the drawing, which can take a significant amount of time for large drawings.

### Drawings

<table>
<thead>
<tr>
<th>Identity column</th>
<th>Allows customizing the name of the identity column. <strong>OID</strong> by default.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Geometry column</td>
<td>Allows customizing the name of the geometry column. <strong>Geometry</strong> by default.</td>
</tr>
<tr>
<td>Version column</td>
<td>Allows customizing the name of the version column. <strong>Version</strong> by default.</td>
</tr>
<tr>
<td>Do not reproject data</td>
<td>Off by default. Check to prevent re-projection of data on export, as might be required to match the projection in use by Manifold to an available Oracle projection.</td>
</tr>
<tr>
<td>Create spatial index</td>
<td>Checked by default. Creates an Oracle spatial index for the table containing the drawing, which makes certain Oracle spatial operations faster.</td>
</tr>
<tr>
<td>Create sequence and triggers</td>
<td>Checked by default. Creates a sequence object to generate values for the ID column, a trigger to fill values in the ID column for newly inserted records, and another trigger to update values in the Version column for</td>
</tr>
</tbody>
</table>
changed records.

**Images and surfaces**

- **Do not reproject data**
  Off by default. Check to prevent re-projection of data on export, as might be required to match the projection in use by Manifold to an available Oracle projection.

- **Create pyramids**
  Create intermediate level images to enable much faster zooming and panning. Always leave this checked unless you are a maximum Oracle GeoRaster expert and have some specialized reason not to create pyramids.

- **Split data into tiles**
  Off by default. If checked, will split image data into tiles of the specified size instead of sending image data in one chunk. For use by Oracle experts who wish to tune image storage in the DBMS.

- **Preserve image library tile structure**
  On by default and used when exporting an image library image to Oracle. When on, the export process will create a separate database tile for each file in the image library, scaling pixels as necessary. When off, the export will send the image library as an ordinary image component and will cut pixels into new tiles if need be.

The default name of the identity column used when exporting a drawing to an Oracle database is **OID** to avoid naming collisions with the **ID** column that stores Manifold object IDs in Manifold drawings.

When we press the **OK** button to tell the **Batch Export** dialog to go to work exporting components that have been checked in the components pane, as each component is exported the box for it will be automatically unchecked. This facilitates canceling the **Batch Export** process, which can be lengthy in the case of many components or large components and then restarting the **Batch Export** process at a more convenient time with only the remaining components checked.

**See Also**

- Database Administrator Edition
- Example: Storing a Drawing in Oracle
- Example: Storing an Image in Oracle
- Export Drawing - Oracle
- Export Image - Oracle
- Images
- Image Types
- Installing Oracle
- Intermediate Levels and Pyramids
- Oracle Express Edition
- Oracle Spatial Facilities
- Tools - Administrator Console
Tools - Customize

The Customize dialog adds and removes panes and toolbars from the main Manifold display. Check the box to show the associated pane or toolbar.

Toolbars will automatically become enabled or disabled for usage as different windows become active. For example, the Format toolbar will be active when we click into a drawing window or into a map window for a map that includes drawings, but it will be inactive when we click into an image window [formatting applies only to drawings].

Therefore, to list a toolbar in the Customize dialog so that it may be checked on or off, first open a window that would normally use that toolbar. It will then be listed in the Customize dialog.
Tools - Options

The **Tools - Options** dialog allows setting of user preferences. Click on an options category in the left-hand pane and settable options for that category will appear in the right-hand pane.

Some options (such as the Location value for File Locations) may be set by double clicking into the option field, entering text, and then pressing **Enter**.

**Automatic Updates**

Configures Manifold to check for updates when it starts up. On by default, with a check every seven days. This option functions only when Manifold is launched interactively. It does not apply to runtime editions, nor does it function if Manifold is launched programmatically, such as within an IMS application.

- **Check for updates automatically**
  - When Manifold starts up, connect via Internet to the Manifold update server and see if an update has been published for the currently running version. On by default.

- **Check every time the application starts**
  - Check for updates every time Manifold is launched interactively.

- **Check every [ ] day(s)**
  - Check only every given number of days, seven by default.

When this function runs, upon startup Manifold will connect to the **manifold.net** update server, check to see if there have been any updates issued for the currently running version of Manifold and, if so, will pop open a message inviting the operator to visit the update site.

No information is ever transmitted to the update server and no data except information on published updates is ever downloaded from the update server.

This command requires Internet access for Manifold using ordinary HTTP protocol, the same as used by web browsers like Internet Explorer. If the computer is not connected to Internet or if HTTP is blocked by a firewall or other security mechanism, this command cannot work. In general, if you can launch a web browser and visit web sites from your computer this command will work.

**Colors**

Specify default colors for formats and user interface items.

**Color Palettes**

Lists color palettes available within the system. Palettes are used for thematic formatting and to color surfaces and terrains. See Customization for information on adding new palettes.

**Confirmations**

Specify which confirmation dialogs will pop up on given actions. Most delete operations, for example, will require a confirmation by default.

- **Confirm bulk operations on editing conflicts**
  - Used with multi-user editing operations. Turns on confirmations for the **Use All Local** and **Use All Remote** toolbar buttons in the Review pane on and off.

- **Confirm closing project containing checked out components**
  - Used with **Enterprise Edition**. Raise a confirmation dialog if a project contains components checked out for editing from the Enterprise server. Until such a component is checked in no one else can check out the component for editing.
Confirm creating undo actions larger than...

Never create undo actions larger than...

Undo functionality for large images and surfaces requires creation of potentially very large temporary files on hard disk. These two settings allow setting limits on what size actions can be undone. The larger the sizes specified the slower the system will perform due to the overhead of making ready for an Undo, if commanded. See the Edit - Undo / Redo topic for more information.

Confirm deleting components

Raise a confirmation dialog before deleting a component.

Confirm deleting drawing objects

Raise a confirmation dialog before deleting an object in a drawing.

Confirm deleting image pixels

Raise a confirmation dialog before deleting pixels in an image.

Confirm deleting labels

Raise a confirmation dialog before deleting labels in a labels component.

Confirm deleting layout objects

Raise a confirmation dialog before deleting an element in a layout.

Confirm deleting table columns

Raise a confirmation dialog before deleting a column (field) from a table.

Confirm deleting table records

Raise a confirmation dialog before deleting a record (row) in a table.

Confirm expanding images or surfaces when pasting

When pasting pixels into an image or surface the image or surface rectangle may need to be increased to house both the original pixels and the new pixels. Confirm before doing so. This confirmation is an important step in case what were thought to be identically georegistered images are in fact located far from each other and thus would result in an immensely large image.

Confirm executing action queries

Raise a confirmation dialog before running an action query.

Confirm exiting application

Raise a confirmation dialog before closing Manifold System.

Confirm modifying table structure

Raise a confirmation dialog before modifying the structure of a table, such as through the Design dialog.

Confirm projecting data

Raise a confirmation dialog before re-projecting data. On by default.

Confirm re-assigning projection after projecting data

Checked by default to warn against the almost always inappropriate use of Assign Projection after Change Projection has been used. Un-checking this option is almost certainly a mistake except for experts in rare and unusual circumstances. See the Inadvertent Usage section in the Edit - Assign Projection topic.

File Locations

Locations to be used by default for files can be specified in File Locations. Locations that may be specified include:

Configuration

The location of customization files.

Also, running a .NET script first searches for referenced assemblies that use relative paths in the Manifold installation folder and then searches...
in the Configuration folder

**Data Cache**
Location of cached files used when linking images from OGC WMS image servers and TerraServer and other image servers when the Cache data between sessions option is turned ON (the default). The default value for the location is `%MyDocuments%`. See the Linked Images from OGC WMS Servers topic, the Linked Images from TerraServer topic and the Managing Cache Files topic.

**Datum Grids**
The location of .gsb and their accompanying .xml files used to specify custom datum grids for NTv2 transformations. By default, contains the value `%Manifold%\Grids` to use a Grids folder within the main Manifold installation folder. See the Custom Datum Grids for NTv2 topic.

**Geocoding Database**
The location of the Manifold Geocoding Data product to use for street address geocoding. See the Street Address Geocoding topic.

**Geocoding Extensions**
Location of the file used for user extensions to the street address geocoding database. See the Street Address Geocoding topic.

**Import and Export**
The location of files used for import and export. Normally specified as a dot, the current folder.

**MrSID Decode Utility**
The name of the auxiliary program used to decode MrSID files into standard GIS format, which must be in the Windows PATH for executable files. By default, `mrsiddecode.exe`. See the Import Image - SID, MrSID topic for details.

**MrSID Info Utility**
The name of the auxiliary program used to provide information on MrSID files, which must be in the Windows PATH for executable files. By default, `mrsidinfo.exe`. See the Import Image - SID, MrSID topic for details.

**Projects**
The default location to use for opening and saving projects.

**Web Pages**
Location to which web sites created with the File - Export - Web Page command should be saved. By default, `C:\Inetpub\Wwwroot`. See the Map Server Overview topic.

In addition to literal settings such as "c:\storage\gisfiles" file locations can be specified using escape sequences as follows:

- `%CommonDesktop%` Desktop shared by all users
- `%CommonDocuments%` Folder for documents shared by all users
- `%Desktop%` User-specific desktop
- `%MyDocuments%` User-specific folder for generic documents
- `%MyMusic%` User-specific folder for music files
- `%MyPictures%` User-specific folder for image files
- `%Manifold%` Manifold installation folder
- `. (dot) Current folder.
For example,

<table>
<thead>
<tr>
<th>Name</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Projects</td>
<td>%MyDocuments%</td>
</tr>
<tr>
<td>Import and Export</td>
<td></td>
</tr>
</tbody>
</table>

...will open the My Documents folder whenever a project is saved or loaded. Import and export will occur in whatever folder was last active.

The above setting assumes we have created a folder called Manifold Projects in the My Documents folder for this user. Opening or saving any project will occur by default within the Manifold Projects folder.

Escape sequences are case-sensitive. %MyDocuments% is not the same as %mydocuments%.

Caution: Cache files persist forever and, in extreme cases, use up your free disk space unless they are manually deleted. See the recommended actions in the Managing Cache Files topic.

Fonts

Default fonts used by various system elements. Fonts used with script windows will be restricted to fixed-width fonts.

Geocoding

Use default country
Set the country to be used if the table does not contain any explicit country field. By default, the United States. Not significant unless MapPoint is being used to geocode addresses outside of the United States. Note: although not all countries listed in the combo box are supported by MapPoint, the list of countries is taken from the Microsoft MapPoint programmatic interface for compatibility with any future extensions that may be offered. This option may be turned off to aid geocoding with both MapPoint North America and MapPoint Europe at the same time.

Use Manifold Geocoding Database
Enabled if the Manifold Geocoding Data product has been installed. Check this box to use the geocoding database (for US street address geocoding). Will be disabled if the Manifold Geocoding Data product is not installed.

Use MapPoint North America
Enabled if Microsoft MapPoint North American edition has been installed. Check this box to use MapPoint’s North American database. Will be disabled if MapPoint North America is not installed.

Use MapPoint Europe
Enabled if Microsoft MapPoint European edition has been installed. Check this box to use MapPoint’s European database. Will be disabled if MapPoint Europe is not installed.

Use geocoding extensions
If checked, the geocoding engine first searches data in user supplied extensions to the street address database and only if a match is not found
does it search the Manifold geocoding data (if installed) and then MapPoint (if used). By default, the option is turned on. See the Street Address Geocoding topic. Will be disabled if no geocoding extensions have been installed.

**Give MapPoint priority over Manifold Geocoding Database**
Check to use Microsoft MapPoint’s database to search for addresses before searching for addresses in the Manifold Geocoding Data product. If checked, Manifold attempts to find a street address in MapPoint and if a building-level match is not found searches the Manifold Geocoding Data product, if installed. Using the default choice of an unchecked box, Manifold attempts to find an address in the Manifold Geocoding Data product first and only checks the MapPoint geocoder if a building-level match is not found. See the Geocoding with MapPoint topic.

**Use the following Manifold Geocoding Servers**
A list of installed modules that support the Manifold Geocoding Server interface for generic geocoding data sources. Check the boxes to enable use of desired modules.

### Import and Export

**Detect read-only data sources for linked components**
Automatically check permissions on the source table of a linked component and make the component read-only if the source table can not be modified. On by default.

**Import data to separate folders**
Create a new folder in the project pane when importing components. Useful when importing complex components that result in the creation of many files.

**Import empty columns in tables**
When importing tables, import even those table columns that contain no values. Off by default.

**Show system tables and views in external data sources**
Show system tables (such as those embedded within Access database .mdb files) when browsing tables in databases. Most database applications hide system tables. Not checked by default.

**Trim imported strings**
Delete leading and trailing token separator characters from imported table strings.

The **Trim strings** option (just like the Trim table transform operators) removes characters listed in the Tool Properties pane’s list of token separators. By default, these are the “white space” characters consisting of the space character, tab, newline and carriage return. Note that adding any other characters to the separator list in Tool Properties will subject them to removal as well if they occur as leading or trailing characters.

### Info Bar

When a new component (other than a new map component) is opened for the first time, by default Manifold can show an info bar in the window reminding the user to verify the projection of that component. The info bar appears whether the component is opened for the first time in its own window or if it is opened for the first time as a layer in a map.

Clicking the info bar launches the Edit - Assign Projection dialog. To verify the projection without any changes simply click OK. If the projection should be changed, make the changes in the Assign Projection dialog and then click OK.

**Prompt to verify projections of new components**
Show an info bar when a new component is opened reminding the user to verify the projection of this component. Checked by default. Un-
Suppress prompt for non-default projections

Show the info bar only if the component has been imported in Orthographic projection or if it has been imported in Latitude / Longitude with coordinate locations outside the expected range (+/- 90 latitude and +/- 180 longitude). Checked by default. Only active if the Prompt to verify projections of new components option has been checked. If the prompt option is on, then un-checking this option will cause the info bar to be shown for all new components.

When Manifold imports components from formats that do not provide coordinate system (projection) information, the component will be imported using Orthographic projection by default. Therefore, the Suppress prompt for non-default projections option, if checked, is a handy way of assuring that the info bar prompt (which rapidly becomes annoying to experienced users if raised for all new components) is shown only when a new component has been imported from a format that does not store projections.

Latitude / Longitude coordinate values outside the expected range (+/- 90 latitude and +/- 180 longitude) are also a tip-off that projected data has been imported from a format that does not store projections, so Manifold will raise the info bar by default in those cases as well.

The default settings are preferred by most users, but they do not guarantee the info bar is raised only when it is required, nor do they guarantee that the info bar will always appear when it is needed. In general, because components imported from geographically-unaware formats will appear in Orthographic projection or when latitudes and longitudes are out of range the short hand rule of urging users to take a look and verifying the projection is a good rule of thumb. But it is not perfect.

For example, the info bar also will be raised when importing components from geographically-aware formats if the component happens to use Orthographic projection. But such cases usually are only a small subset of all projected components that are imported.

A more serious situation is that sometimes certain formats like shapefiles accompanied by .prj files or image files accompanied by "world" files can give enough indication of a coordinate system to cause a projection other than Orthographic to be assigned but not enough coordinate information to have all parameters assigned correctly. In that case, it could be useful to have the info bar raised to remind us to check the assigned projection in all details, but the default options settings will not cause it to be raised.

If new users will be doing a lot of imports from projected shapefiles or from formats that use "world" files, it may be a wise choice to un-check the Suppress prompt for non-default projections option. This will cause the info bar to be raised on all new components, causing some annoyance but also reminding the new users to always verify the projection of a newly-imported component.

Logging

Manifold can automatically log information to the History pane when the following options have been checked:

- **Log query execution time**
  Logs execution time of queries.

- **Log rendering time**
  Logs the time required to render a given window.

- **Log script execution time**
  Logs execution time of scripts.

- **Log transform time**
  Logs execution time for the Surface - Transform dialog, spatial overlays and topology overlays in addition to Transform Toolbar operations.

Command line switches take priority over the above settings if Manifold is launched using one or more command line options for the above.

Why have options to log the time required for something like rendering? Developers who create applications based upon Manifold are often acutely interested in creating applications that render as rapidly as possible.
They'll use timings to explore how altering different factors like those mentioned in the Performance Tips topic will affect performance.

### Miscellaneous

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smooth freeforms at Rounding radius to use to include / exclude pixels when working with images using tools like Paint Freeform or Select Freeform.</td>
<td></td>
</tr>
<tr>
<td>Limit number of sequential approximations</td>
<td>Some computations within Manifold use a series of sequential approximations. This number gives the number of sequential approximations to compute.</td>
</tr>
<tr>
<td>Automatically list completion suggestions when editing queries</td>
<td>Automatically popup lists of possible completions when editing queries. See the Autocompletion in Queries topic.</td>
</tr>
<tr>
<td>Automatically select query records</td>
<td>When querying a drawing, automatically deselect any selected objects and then select all objects returned by the query. When a query includes more than one drawing, deselect all previously selected objects in all of the drawings and then select objects in the first drawing for which an ID column is found in the query. On by default.</td>
</tr>
<tr>
<td>Compress .map files to save space</td>
<td>On by default. Will compress a .map file using &quot;zip&quot; style compression when creating or saving a .map file.</td>
</tr>
<tr>
<td>Create empty document on startup</td>
<td>Creates a new, empty project when Manifold is launched.</td>
</tr>
<tr>
<td>Fill new pixels with current color</td>
<td>If checked, uses foreground color for new pixels added with the Add Margin transform operator. If not checked, uses transparent color for new pixels.</td>
</tr>
<tr>
<td>Make new queries ANSI-compatible</td>
<td>Use ANSI-compatible SQL to parse any new queries. Checking this box will set the ANSI compatibility option in any new queries. See the Queries topic.</td>
</tr>
<tr>
<td>Make new queries cached</td>
<td>Set the Use cache option on/off by default for new queries. See the Queries topic.</td>
</tr>
<tr>
<td>Override system data with custom config files</td>
<td>Off by default. When turned on, the system allows user-supplied configuration entities (for datums, ellipsoids, palettes, presets, styles and units) to override corresponding system entities with the same names. When turned off, the system rejects user-supplied configuration entities whose names are the same as system entities. Toggling the value of this option requires a restart of Manifold.</td>
</tr>
<tr>
<td>Process global events with scripts</td>
<td>Allows (the default) or disallows the processing of click and double click events with scripts. See the Click Events topic.</td>
</tr>
<tr>
<td>Refresh linked components after opening file</td>
<td>Turned on by default. Fetch current data for any linked tables, inked drawings, linked images or linked surfaces when a .map project file is opened. This option has effect even if a script is used to load a .map project that contains linked components. The names of any linked components that failed to refresh in the process of opening the .map file will be logged into the History pane.</td>
</tr>
<tr>
<td>Save file prior to running script</td>
<td>Off by default. A safety measure to save the .map project file before running a script.</td>
</tr>
</tbody>
</table>
Set transfer rules for new columns to Copy / Copy
If checked, changes the default behavior of transfer rules for newly created columns from Sample to Copy for both 1 to N and N to 1 transfers.

Shade new surfaces
Turns on Shading option in the View - Display Options dialog for new surfaces. On by default.

Split branched objects after transforms
Automatically decompose branched objects into separate non-branched objects after running a transform toolbar operator.

Use custom datum transformations
Use NADCON formulae and similar custom transformation methods when converting between datums in locations served by such custom methods.

Use English measurement units
Toggles between Metric units and English units. If checked, use English units.

Use GDI-compatible offscreen cache
Off by default. This option uses a lower performance method of rendering the display screen that may be used in case of buggy or incomplete video drivers. For example, some ATI-based cards have a driver bug that causes the drawing display to refresh and then go blank. To avoid triggering the ATI video bug, check this option. A better solution, of course, is to use a video card that does not have buggy drivers.

Use GPGPU technologies (NVIDIA CUDA)
On by default. Enables use of NVIDIA CUDA massively parallel processing if appropriate CUDA-capable hardware and CUDA drivers have been installed. See the NVIDIA CUDA topic.

Printing
Optimize printing of images and surfaces
When on, the system avoids rendering image and surface components in full resolution during print job setup, deferring final rendering to the printer driver, provided the image or surface can be rendered with either no re-projection or with simple scaling and shifting. This makes the size of the print job smaller and offloads the task of scaling raster images to the printer driver. Most printer drivers can automatically re-scale images and surfaces within the driver for better performance and reduced print job size. Some drivers cannot do so and will so cause the job to be printed more slowly than if Manifold renders images and surfaces. By default the option is turned on since most modern printer drivers can automatically render and re-scale. If the print job is slow or if the printer or plotter in use is suspected to be Windows unaware, try turning this option off and see if performance improves.

Print blank pages in multipage printouts
Print all pages in multipage layouts, even those that may not have any ink on them. See the Multipage Layouts topic.

Print using GDI+
On by default. If unchecked, prints using GDI mode instead of GDI+ mode. GDI is an earlier Microsoft technology. Provided for use in case of a buggy or incomplete printer driver. Printing in GDI mode does not support antialiasing or transparency for drawings. See the Layer Opacity topic for information on transparency. Curved (rotated) or labels aligned to lines cannot be printed n Windows 9x systems if the Print using GDI+ option is turned off.
Proxy Server

Allows specifying parameters for a proxy server to use while retrieving data from the Internet, for example, when using images linked from OGC WMS servers.

Connect via proxy server
Check if a proxy server is being used.

Address
URL for proxy server.

User
User name if required for proxy server.

Password
Password if required for proxy server.

Different proxy server settings may be required for different tasks. For example, at a given domain that uses a proxy at proxy.mydomain.edu a user connecting to an ECWP server might use a setting of:

http://proxy.mydomain.edu

…but for a WMS link it may be necessary to include the port:

http://proxy.mydomain.edu:8080

This depends upon the proxy server in use and how it is configured and the connection to be achieved.

Rendering

Options that control rendering of text and graphics objects, both onscreen and in certain rendered exports such as exporting layouts to AI, PDF or PS files.

Adjust display scale for monitor resolution
Automatically scale objects so that size 1 objects are 1/72 inch (one point) regardless of how many pixels that requires in the monitor resolution in use. This option also automatically scales text to preserve desired point size within component windows regardless of the Windows Display setting for Large Font / Small Font. This option also provides the same physical representation on very high resolution monitors.

Antialias lines
Use antialiasing for a smoother line appearance. On by default. When on this option also antialiasises area style patterns, area borders and the borders of text box labels. See Lines and Line Styles.

Antialias text
Use antialiasing for smoother appearance of text labels. On by default. When on this option will also antialias TrueType symbols in point styles, excepting those using halos.

Render data progressively
Enables progressive rendering so the contents of windows showing large or complex data are rendered in stages, with the system becoming responsive to other commands before rendering is completed. This is very useful with large, complex components that might take some time to render. For example, even before a large drawing finishes rendering we can begin zooming in to a region of interest. When a component is small enough to render very quickly, progressive rendering will not be noticed because the component will be rendered in a single pass. On by default.
Smooth large vector objects for performance

Checked by default. Renders large, complex vector objects with reduced detail. Results in faster rendering performance without significant degradation of visual appearance.

Scripts

Options of interest to scripting:

Cancel user interface operations after ... seconds

Specifies the number of seconds the system should wait before timing out during user interface scripting. 120 seconds by default.

Process global events with scripts

Allow scripts to process global events. On by default.

Save file prior to running scripts

Save the .map project file before running a script. Off by default.

Server Storage

Options related to Enterprise Edition only. If you do not have Manifold System Enterprise Edition installed these options will not be applicable

Cache newly shared components

Specifies whether newly shared components will be cached or uncached. By default, this option is checked so that newly shared components are cached. If this option is not checked, newly shared components will be uncached. See Cached and Uncached Components.

Get latest versions of shared components after opening file

If checked, when a project .map file containing shared components is opened the system will execute a Get Latest Version for each shared component. This will fetch the latest version from the component's Enterprise server if the component has changed since the project has last been opened. Not checked by default.

Refresh state of shared components every [ ] seconds

Specifies time interval on which Manifold will check the Enterprise server for each shared component to see if it has changed status, such as, for example becoming checked out by another user. 120 seconds by default.

Sounds

At times we may launch a very demanding command that will take some time to accomplish. Some jobs may be so computationally intensive that they may require hours or longer to run. It is often convenient to set such jobs running while we continue work on another computer; however, in such cases it is also convenient to have Manifold play a sound or otherwise alert us when the job is done.

The Tools - Options - Sounds dialog allows us to specify whether or not we want a sound to be played at the conclusion of a lengthy process, how long the process must be before a sound is played and, optionally, a .wav file from which a sound can be taken.

Play sound after lengthy operations

Check to enable sound-playing after the completion of a lengthy process.

Wait at least ... seconds

The length of time before a process is considered "lengthy." For example, if this value is set to 30 seconds any operation that takes less than 30 seconds will conclude silently.
Sound file
An optional .wav file to use for sounds. Leaving this box blank or choosing a file other than a .wav file will result in use of the default "beep" sound. Use the Play button next to the browse button to play the specified file to preview how it sounds.

It's fun to browse a Windows system for .wav files to use for sounds. Windows ships by default with many sounds.

Status Bar
Status bar options configure the status bar at the bottom of the Manifold display. Changes take effect immediately. See the Getting Started topic for notes on the status bar.

Component-specific information
Readouts relevant to particular components, such as the number of selected objects (appears for drawings for example, but not for comments components).

Coordinate system
Report the coordinate system (projection) in use.

Current location
Location readout for the mouse cursor using various ways of reporting coordinates. Options with examples beginning with the default are:

- latitude / longitude:
  122°19.701' W 37°34.119' N
- latitude / longitude (decimal):
  -122.3816 37.5597
- latitude / longitude (deg-min-sec):
  122°19'04" W 37°33'25" N
- projected coordinates:
  550833.8339 4155957.9227
- USNG coordinates:

Current modes for mouse, selection and tools
Show mouse status such as snap mode, status of selection mode and the current command being used.

Current scale
Show scale on status bar using Report setting.

Report
Choose style of scale or size status readout. Scale is accurate only for projected components. Options are:

- absolute scale (#:#) - Absolute scale in 1:xxx form.
- relative scale - Relative scale as ratio between one centimeter or inch and the scaled number of meters/kilometers or feet/miles, depending on setting of the Use English measurement units option.
- horizontal extent - Shows the horizontal size of the current window.

Note: The relative scale option will automatically switch from large units to small. For example, in metric units once the scale falls below 1 centimeter to 1 kilometer it will switch to reporting the number of meters per cm.

User Interface

Grid lines in tables
Show horizontal and/or vertical lines in tables to separate rows and columns.

Paint palette colors using []
The size of color well boxes shown in palette
<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pixel boxes</td>
<td>windows for palette images.</td>
</tr>
<tr>
<td>Zoom magnification factor</td>
<td>Specify the change in magnification to use when zooming in or out in the range from 1.1 to 4.0. The default value is 2.0.</td>
</tr>
<tr>
<td>Apply table cell editing to all selected records</td>
<td>On by default: editing one cell in a set of selected records will apply that value to all such cells in the selected records.</td>
</tr>
<tr>
<td>Automatically set insert mode</td>
<td>When editing drawings, automatically choose Create Areas, Create Lines or Create Points depending on which shape creation tool is used. See Adding Shapes.</td>
</tr>
<tr>
<td>Automatically set paint mode</td>
<td>When editing images, automatically choose Paint Areas, Paint Lines or Paint Points depending on which shape creation tool is used. See Image Editing Tools.</td>
</tr>
<tr>
<td>Autoscroll window on edit or selection operations</td>
<td>If the mouse is in a command mode (such as drawing a selection box) automatically pan (scroll) the window when the mouse nears the edge of the window. Indispensable in some operations and insufferable in others if inappropriate options are specified. On by default.</td>
</tr>
<tr>
<td>Margins</td>
<td>Gives the distance in pixels from the edge of the window at which autoscroll occurs. Set to 5 pixels by default.</td>
</tr>
<tr>
<td>Distance</td>
<td>The distance in pixels the window is scrolled for each autoscroll. Set to 10 pixels by default.</td>
</tr>
<tr>
<td>Require Alt key</td>
<td>Disables autoscroll unless the Alt key is pressed. Off by default.</td>
</tr>
<tr>
<td>Highlight entire line in script window</td>
<td>Changes the background color of script lines containing bookmarks, breakpoints or the current execution line. Breakpoint colors override bookmark colors, so if a line contains both a bookmark and a breakpoint it will be highlighted with the breakpoint color. The current execution color overrides both breakpoint and bookmark colors.</td>
</tr>
<tr>
<td>Open newly created component</td>
<td>When a new component is created in the project pane, automatically open it in a window.</td>
</tr>
<tr>
<td>Pop up Project pane when creating or opening a file.</td>
<td>Automatically pops open the Project pane when creating or opening a component.</td>
</tr>
<tr>
<td>Quiet system activity indicator</td>
<td>On by default. Causes the system activity indicator in the lower right corner of the display to show a 3D “bump” when the system is busy. Unchecking this option causes the system activity to toggle from a large green dot to a red dot when the system is busy.</td>
</tr>
<tr>
<td>Restrict mouse wheel to only operate over focused window</td>
<td>On by default. Disables the wheel in wheel mice when the mouse pointer is not over the window that has the focus. This prevents accidentally changing the current scale via the scale combo box in the Navigation toolbar and other unintentional actions using the mouse wheel.</td>
</tr>
<tr>
<td>Save last used view in component</td>
<td>If checked, zoom to a view called Startup in the views pane if such a view exists, or to the most recently used view when opening a drawing, image, labels, map, profile, surface or theme window. On by default. If this option is not checked, then on opening the component will be zoomed to fit the display window.</td>
</tr>
</tbody>
</table>
Show grid lines in Variables and Watches: Show horizontal and vertical lines in these panes for greater visual separation of cells. Off by default.

Show splash screen: Display the Manifold System splash screen when Manifold launches.

Use Explorer-style menus and toolbars: Removes all colors from toolbar buttons until mouse hovers over them as in IE. Since Manifold does not use gaudy icons like IE the effect is bland and not recommended. Takes effect the next time Manifold is launched.

Use selection style for palette: Use defined selection style to show which palette box is chosen in a palette window used with palette images. If not checked, a plain border style will be used.

Image Menu

Image Menu

The image menu appears when the focus is on an image window or an image layer is active in a map. Commands listed below will appear either under the main Image menu or under submenus such as Adjust or Effects. Commands are enabled for RGB or RGBA images. To use these commands with other types of images, first convert them to RGB or RGBA images.

Open Palette: Enabled for palette images. Open the palette for this image.

Dither: Uses a smaller number of colors in an image while preserving a perceptual effect through the use of dither patterns.

Quantize: Convert image to the specified number of colors using enhanced or standard methods.

Resize: Change the size of the image using desired interpolation method.

Convert To: Convert images between grayscale, palette, RGB or RGBA images.

Adjust Menu

Color Balance: Adjust RGB color values for Shadows, Midtones or Highlights.

Brightness / Contrast: Change brightness and contrast.

Hue / Saturation: Alters hue, saturation and lightness.

Invert: Invert pixel channel value. Creates a "photographic negative".

Equalize: Alters individual pixel brightness so there is an equal number of pixels at all brightness levels.

Gamma: Change mid-tone color brightness.

Threshold: Forces grayscale pixels to white, black or unchanged based on a histogram.

Threshold Color: Like Threshold but applies to each RGB channel individually.

Posterize: Convert image to a limited number of colors for a "poster"
Effects Menu

**Colorize**
Adjust saturation and hue while keeping intensity the same, as if a color were applied to a grayscale version of the image.

**Gaussian Blur**
Apply Gaussian bell curve blurring to create shadows, halos and transitions.

**Motion Blur**
Streaked effect in the desired direction.

**Diffuse**
Moves pixels by swapping them in a "random walk."

**Filter**
Applies convolution filters, either custom filters or a choice from numerous presets.

**Fluoresce**
Alter saturation and lightness in asymmetric ways to cause some colors to appear unusually bright.

**Noise**
Add random monochromatic or color pixels.

**Relief**
Add highlights and shadows to give a 3D effect to 2D images.

**Simplify**
Aggregates pixels into larger clumps for a pointillist or mosaic tile effect. Can also remove small dots in cartographic work.

**Tile**
Convert image into rectangular tiled regions of appropriate color.
Image - Brightness / Contrast

Allows alteration of brightness and contrast of selected pixels or over the entire RGB or RGBA image if no pixels have been selected. **Brightness** makes the image lighter or darker overall, while **Contrast** either emphasizes or de-emphasizes the difference between lighter and darker regions. See the Colors as Hue, Saturation and Brightness topic for more information on brightness.

- **Brightness** Increase or decrease the brightness of pixels. Low brightness will result in dark tones while high brightness will result in lighter, pastel tones.

- **Contrast** Increase or decrease contrast. Increasing contrast increases the apparent difference in lightness between lighter and darker pixels.

- **Preview** Check to see effect in action.

These commands are also available on the Transform Toolbar for images as the **Brightness** and **Contrast** transform operators. However, when used from the transform toolbar the effects are immediate with no preview possible.
**Image - Colorize**

Adjusts hue and saturation while keeping intensity the same in RGB or RGBA images. The resultant effect will appear to apply the designated color to a grayscale version of the image.

<table>
<thead>
<tr>
<th><strong>Preset</strong></th>
<th>Choose color to apply by name from a list.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Apply</strong></td>
<td>Apply the RGB settings for the effect in the preset box to the slider bars and RGB values.</td>
</tr>
<tr>
<td><strong>Save As...</strong></td>
<td>Save current RGB setting as a preset color.</td>
</tr>
<tr>
<td><strong>Delete</strong></td>
<td>Delete previously-saved preset color or disable factory preset.</td>
</tr>
<tr>
<td><strong>Red</strong></td>
<td>Colorize value for Red channel.</td>
</tr>
<tr>
<td><strong>Green</strong></td>
<td>Colorize value for Green channel.</td>
</tr>
<tr>
<td><strong>Blue</strong></td>
<td>Colorize value for Blue channel.</td>
</tr>
<tr>
<td><strong>Preview</strong></td>
<td>Check to see effect in action.</td>
</tr>
</tbody>
</table>

**Preset** colors allow choice of a standard range of named colors, as well as saving one's own colors as presets. Because there are many presets and because at times very large images may be manipulated in Manifold, choosing a preset does not apply that preset's color values to the slider bars until the **Apply** button is pressed. This makes it possible to browse the presets without applying the color values to the main image regardless of whether the **Preview** box is checked or not.

The selected area of the image (shown using **Border** selection style) has been colorized with the **Dark Orchid** preset color.

Colorize is often used to add color tones to gray scale images. The image should first be converted to an **RGB** image using **Edit - Convert to** if it is not already an RGB image.

Suppose we wish to colorize a circular region in this overhead view of downtown Palo Alto near the 525 University Avenue office building. First, we select the circular region (shown using **Border** selection style). If we did not select a region the colorize effect would be applied to the entire image.
We click the **preview** box and move the **Green** slider to the right to increase green values and the **Red** and **Blue** sliders to the left to decrease red and blue values. The result is colorization into green tones.

The final result is a convenient means of highlighting a particular region within an image.

A more sophisticated effect may be achieved by choosing **Edit - Select Inverse** to select all the pixels **not** in the original circle, and to then use the **Brightness / Contrast** control to increase brightness while decreasing contrast slightly. This will "wash out" the rest of the image and make the round green region stand out even more.

**Colorize and All-Black Images**

Because colorize keeps intensity the same attempting to colorize an all-black image will result in no change. Black is zero intensity for all color channels. To use colorize to change the color of a solid color layer it must be some visible shade of gray or white.
Image - Color Balance
Adjusts RGB color values in an RGB or RGBA image for Shadows, Midtones or Highlights.

Tones Choose from Highlights, Midtones or Shadows

Red Value for Red channel.
Green Value for Green channel.
Blue Value for Blue channel.

Preserve Luminosity Adjust all three channels slightly to preserve perceived lightness / darkness.
Preview Check to see effect in action.

Select the tonal range to be affected and adjust the slider. Switching tonal ranges in the choice box will preserve any changes made in the other tonal ranges. For example, lowering the Green value with the choice box set to Midtones, and then switching to Shadows in the choice box will still have the Midtones value changed to the lower value, as can be seen by switching back to Midtones. The tonal range refers to the brightness of colors.

In complex color images,

- Moving the Red slider left and right will appear to change the tonal variation from Cyan to Red tones.
- Moving the Green slider left and right will appear to change the tonal variation from Magenta to Green tones.
- Moving the Blue slider left and right will appear to change the tonal variation from Yellow to Blue tones.

The above effects occur because, for example, turning Blue color all the way down leaves only Red and Green tones which make up Yellow. See the discussion of Channels in the main Images topic.

The above illustrations show the original image in the center with Midtones in the choice box and the Blue slider moved to the left for the left image and moved to the right for the right image. Note that the increasing blue to the right or decreasing blue to the left ("increasing yellow") does not affect the white clouds, which are "highlights" or dark shadows as much as it does the middle range of color tones.
Image - Create Index Drawing

When an image library is open the Image - Create Index Drawing command is enabled. This command allows us to automatically create a drawing that contains areas showing the extent of each individual image file that makes up that image library together with desired information about each image file, such as the name of the file, the path to that file's location and the type of file.

Index drawings are used in a variety of applications. For example, we might not wish to display all of the images in an image library but need to fetch an image that is part of the library. An IMS application, for example, might show a variety of vector layers and use an index drawing to show where imagery is available in a particular region. The application could be programmed so that a user who wants to see that imagery can click within an index drawing area to pop open the corresponding image in a new window or to make it visible in a map window.

To create an index drawing for an image library:

1. Open the image library.
2. Choose Image - Create Index Drawing.
3. Provide a name and an optional description for the new drawing.
4. Choose which columns should be added to the drawing's table. Press OK.

### Controls

- **Name** The name of the new index drawing.
- **Description** An optional description to appear in the drawing's properties.
- **Select All** - Check all columns.
- **Select None** - Uncheck all columns.
- **Select Inverse** - Uncheck all checked columns and check all unchecked columns. A fast way to use all but one column: click Select None, check the one column not desired and then click Select Inverse.
- **Columns** Check which fields are to appear as columns in the drawing's table. The columns available will depend upon the method used to assemble the image library. The
The illustration above shows columns available when assembling an image library using image tile file names to position tiles.

**Include autogenerated images**  
Off by default. Checking this box will add intermediate level images if these have been automatically generated for this image library.

**Example**

Let's create an index drawing for the image library used as an example in the Image Libraries topic.

The image library is called **Bay image** and is assembled from nine .tif format files using accompanying .xml files to position the tiles.

It appears in our project pane as a linked image. The first illustration above shows the image library as it appears in a **Map** component.
With the focus on the opened map window, we choose *Create Index Drawing* in the *Image* menu. The *Create Index Drawing* dialog opens. The *Name* will be filled in for us with a default suggestion for the name of the new drawing derived from the name of the image library. We will use this name and choose other options as seen above. Press *OK*.

The result is that a new drawing appears in the project pane.

If we click open the drawing's table we can see it has nine records in it, one for each area object, that show the extents of the images that make up the image library for which the index drawing was created. The columns that appear in the drawing's table are the columns we selected in the *Create Index Drawing* dialog.

To see what the drawing looks like, we can drag and drop it from the project pane into the open map window.
Our first impression is somewhat underwhelming: the index drawing is formatted using default drawing formatting so each area in the drawing has an opaque background. We can't see anything behind this drawing because of the opaque color of the area backgrounds.

That takes but a second to change by using the format toolbar to change the formatting of area background to transparent color.

While we are at it, we'll change the area border color to something bright and visible, and we will add a labels layer that automatically creates labels from the Filename column.
The end result is a map that shows a labels layer and our new index drawing above the image library layer. The rectangular areas in the index drawing (now seen only by their borders) show the extents of each of the nine image tiles that make up the image library. The labels layer shows the name of each tile.

**IMS Example**

Once we have information about images within an index drawing we can proceed to do all sorts of wonderful things. Suppose, for example, that we have a Manifold IMS web site at [http://www.mydomain.com](http://www.mydomain.com) and we would like visitors to be able to click open an image by clicking into a map.

We can open the drawing’s table and then use Manifold’s ever-useful **Edit - Replace** command to edit the path into an appropriate URL path. (Experienced IMS players know already where we are going with this example...)

We then change the type of the column to **URL** and we also name this column **URL** so that it can be recognized by the **Hyperlinks** option in the **Export Web Page** dialog we will use when exporting the project as an Internet Map Server web page. The **Hyperlinks** option gives an IMS web page template the ability to automatically launch a hyperlink when an object containing that hyperlink is clicked.

We take the images that made up the image library and we copy them into an **images\bay** folder within the **InetPub\wwwroot** website folder on the web server that runs our **mydomain.com** web site. Because we have had the wisdom to read the IMS topics in this documentation, we also know to make sure that the **IUSR_** account for the web server has read access to those image files as well as read access to the folders that contain them.
The IMS site we create has a variety of drawing layers in it with the index drawing at the top, now with borders formatted in a pretty green color. None of the layers are clickable (set these properties in the layer restrictions for the map component) except the index drawing layer. In this example we do not show the image library as a layer.

Note that even though the areas that comprise the index drawing have been formatted with transparent background color, those areas are still there. Clicking within the borders of the rectangular areas is still a click onto that area object even though it may be transparent.

Whenever someone clicks onto one of the index drawing areas, Manifold IMS will hop to attention, grab the corresponding URL hyperlink and in a flash launch that URL within a new browser window or other program, as appropriate, to display that image.

It’s true that our use of .tif images for this IMS example is not the most intelligent thing, since most browsers will not immediately display a .tif but instead are configured to launch some image viewing application whenever they are fed a URL that ends in a .tif file. For a “real world” web site we might have preferred to use an image format such as .jpg or .png that could always be immediately viewed in a new browser window.

Likewise, the default templates in Manifold IMS are not necessarily the best for building websites with hyperlinks in them, since each click not only launches the URL but also (since the Center tool in the default templates is always on by default) will center the display to the position clicked. Many designers prefer to have their sites act that way but if different behavior is desired one can always use the sample web applications published at the manifold.net site as a guide to creating different navigation tools.

The IMS example discussed above is a fairly common use of both image libraries and index drawings. Public utilities, for example, often have in addition to their vector GIS drawings a large amount of legacy information in the form of scanned blueprints and other scanned images. Utilities will often georegister those scanned images so they may be used in a GIS environment. However, because images are so much less efficient to work with than drawings the utility will not use those images except when it is absolutely necessary to view them.

In such cases, a utility might create an image library consisting of the various scanned images it has and then create an index drawing. The index drawing is a fast, “lightweight” layer that can be used in the utility’s GIS work to show where images are available and to provide a convenient way of locating an image when it is required.

See Also

Image Libraries
Intermediate Levels and Pyramids
Labels
Creating Labels from Fields
Image - Diffuse

The Diffuse command moves pixels in RGB or RGBA images by swapping them in a "random walk." Reapplying Diffuse several times will move pixels further and further away from their original positions. Diffuse is used to apply artistic effects. If a selection is present, Diffuse will ignore any unselected pixels. Diffuse will also ignore invisible pixels.

Level  The number of times diffuse should be reapplied.

Preview  Check to see effect in action.

The mages below show a zoomed-in view with one application of Diffuse as well as a view after several applications of Diffuse.

Note how pixels "drift" at random from their original positions with repeated applications of Diffuse.
When used with photographic images, **Diffuse** provides a pixelated, "dry brush" painterly effect. The illustration above shows the effect after using a level of **9**, applied to the sample **bronze** image that was previously altered using the fluoresce command.

If a selection is present, the action of the **Diffuse** command is limited to the selected pixels.

For example, in the image above a circular region was selected and then **Diffuse** was applied.
Image - Hue / Saturation
Allows simultaneous changing of hue, saturation and lightness in RGB or RGBa images. See the Colors as Hue, Saturation and Brightness topic for more information on these concepts.

**Hue**  Shift color of all pixels clockwise or counterclockwise about the color wheel.

**Saturation**  Increase or decrease color saturation. Fully-saturated colors are very bright, while low saturation are grayish.

**Lightness**  Increase or decrease lightness. Zero lightness results in all black pixels, while full lightness is white.

**Preview**  Check to see effect in action.

The **hue** slider shifts the colors of all pixels in a plus or minus direction around the color wheel. Plus or minus 180 degrees are equivalent. Blue tones shifted 180 degrees will end up yellow, as may be seen in the blue sky in the sample pictures. Use **hue** to shift images slightly, such as more or less red or blue throughout the image.

The **saturation** slider increases or decreases the intensity of colors, ranging from white (no saturation) to grayish colors to bright, vivid colors (full saturation).

The **lightness** slider controls how light or dark each pixel as compared to an absolute gray scale from black to white.

Increasing lightness goes from black to colors to white.

See also: Colors as Hue, Saturation and Brightness

**Note:** The illustrations above are shown using only 256 colors and thus will appear "grainy" as compared to true colors in Manifold images.
Image - Dither

This command replaces pixels in an RGB or RGBA image with a "dither" pattern that approximates colors of pixels by using a smaller number of colors in a mixed pattern. The dither colors are selected so that the human eye interpolates between them to get the desired color impression at that location.

Palettes

Adaptive Sample colors appearing most frequently in the image and use these to choose a best fit to the given number of colors from the entire RGB space.

Linear Choose colors that are evenly spaced in the RGB cube space for the given number of colors.

Photoshop Use the 90-color Photoshop swatches palette. Handy for many restricted-color graphics arts images.

System Use the 256-color Microsoft Windows system palette.

System (Mac) Use the 256-color Apple Mac system palette.

VisiBone 2 Use the VisiBone web designers palette.

Web Use standard 216-color palette included by default in all web browsers.

Palettes used by other applications and systems are provided so that images destined for those systems can be created using an exact match of palettes.
The Download command appears when the focus is on a window for an image linked from an external image server such as an OGC WMS image server, a Microsoft TerraServer image server or a Manifold Image Server compatible third party module such as Google. The command allows automatic download of all tiles available for the image at any level of resolution available. Tiles will be assembled into a single image at each resolution level selected.

Note that before the Download command appears, we must have already used File - Image - Link to create a linked image from an image server such as TerraServer or an OGC WMS server. The maximum detail available in the downloaded image will have been set by the choices made when initially linking the image, such as the use of the Min Scale parameter in the TerraServer dialog or the Size parameters in the OGC WMS dialog.

Select All - Check all the boxes in the levels pane.
Select None - Uncheck all the boxes in the levels pane.
Select Inverse - Check all unchecked boxes and uncheck all checked boxes in the levels pane.

(levels pane) Check levels to be downloaded. The Level column reports the resolution in pixels at each level. The Tiles column reports the total number of tiles as well as the approximate number of tiles already linked at each level and thus available from cache, if caching is on (the default) for faster download.

(status bar) The number of tiles to be downloaded of the total number of tiles available with the current set of levels checked. The number to be downloaded is in addition to those already in cache and so provides a useful guide when the connection to the image server is slow.

The image download process will run in background to allow other work to proceed while the image is downloaded. Take care not to capriciously select very detailed levels for download if your connection to the image server is very slow. Downloading levels of an image linked from a remote image server proceeds from the least detailed levels to most detailed levels. This provides a reasonable chance of getting at least some useful, complete levels in case a connection is interrupted.

Images may be downloaded at a variety of detail levels. The most detailed level that may be downloaded will be set by the parameters specified when linking the image (Min scale in the case of TerraServer images and Image size in the case of OGC WMS images). The least detailed level will be set by the least detailed data level available on the server. The Download dialog allows us to choose which level of detail we want between the most detailed and the least detailed available and to download the resulting image into a local image so that subsequent work with the image is faster or more convenient.

For example, we could use the Download dialog to fetch a complete image from the server so we can later work with that image when disconnected from the server. We might also use the Download dialog to download all tiles in the most detailed image level prior to exporting the image to a local file, perhaps as an ECW image.

Image - Download and Image - Relink / Unlink Compared

An image linked from an image server may be converted to a local image within the project either by using the Image - Unlink command or the Image - Download command. The Unlink command converts the existing linked image to a local image at whatever resolution is currently in use. The Download command may be used to automatically fetch all image tiles at any available resolution to create a local image at that resolution.

Images linked from external servers normally may also be exported to image files. In that case, the exported image will be the same as that which would be obtained by unlinking the linked image.

Note that exporting or unlinking an image linked from an image server will only use the data already downloaded from the server and will not attempt to download any more data. It will also only use the data for the most detailed image level. It is usually a good idea to use Image - Download to check how much data has already been downloaded for the most detailed image level prior to exporting or unlinking the image.
Another way to create a local image from a linked image is to open the linked image and then to use the keyboard F6 command or the Tools - Make Image command. The Make Image dialog allows us to create a local image in several different ways from the linked image without unlinking the linked image. For example, we can use this command to create an image of the displayed window or of the entire linked image at a desired number of pixels or scale.

**Very Important**

Before using Export with an image linked from an image server, see the Very Important discussion in the Image - Unlink command.

**See Also**

Linked Images from Google Servers  
Linked Images from Manifold Image Servers  
Linked Images from OGC WMS Servers  
Linked Images from TerraServer  
Image - Relink / Unlink  
Tools - Make Image
Image - Equalize

**Equalize** builds a histogram of colors used in all pixels in the RGB or RGBA image from the brightest to the darkest and then alters colors of pixels in between so there is the same number of pixels at all brightness levels.

**Level**
Offset the brightness level higher or lower from the default 50% level. Increasing level results in higher overall brightness.

**Preview**
Check to see effect in action.

This command is also available on the Transform Toolbar for images as the **Equalize** transform operator. However, when used from the transform toolbar the effect is immediate with no preview possible. In the transform toolbar version the parameter specified is used as the level.

**Example**

Starting with the **schloss** sample image applying an **Equalize** with a level of 0 (seen above) is similar to the Auto Level transform. It provides better balance between light and dark tones to give more character to what started as a slightly washed-out image. Applying **Equalize** with a level of -76 shifts the tones to darker levels for an almost "sunset" effect as seen below.
The **Filter** command works with RGB or RGBA images and exposes the internal functioning of many commands in the Transform toolbar and allows us to add new, custom commands using convolution filters. Most users will apply the commands in the Transform toolbar without ever examining their matrix values in the **Filter** command. However, for those users who are so inclined, the **Filter** command provides an “open door” to altering existing filters and creating new ones. If a selection is present, **Filter** will ignore any unselected pixels. **Filter** will also ignore invisible pixels.

Many image effects are created through a mathematical process called **convolution**, where the brightness values in all channels of each pixel are recomputed based on the values of surrounding pixels. The rules for how each pixel’s brightness values should be computed are given by a matrix of numbers, where the central number gives the brightness of the pixel and the other numbers set forth how values from surrounding pixels should be multiplied into the convolution.

Choosing different numbers for the convolution matrix results in a very wide range of visual effects such as sharpening, blur and so on. Manifold provides a 5 x 5 matrix, where each pixel may be adjusted based on reference to 24 surrounding pixels. This convolution matrix will be applied in turn to each pixel in the image.

**List Box**  Choose a preset filter from a list. Choose **Custom** to create a new filter.

**Apply**  Load the convolution matrix with the numbers for the filter currently shown in the preset box.

**Save As…**  Save current convolution matrix as a preset filter.

**Delete**  Delete previously-saved preset filter or disable factory preset.

**Matrix boxes**  Values for convolution matrix calculation.

**Preview**  Check to see effect in action.

The list box at the top of the dialog allows choice of a filter from many preset filters. Because there are many presets and because at times very large images may be manipulated in Manifold, choosing a preset does not apply that preset’s convolution matrix numbers to the matrix until the **Apply** button is pressed. This makes it possible to browse the presets without applying their convolution matrix to the main image regardless of whether the **Preview** box is checked or not.

The **Filter** dialog has a choice box at the top for preset filter effects. Choosing **Custom** allows us to enter our own values to create custom filters. In addition to appearing in the **Filter** dialog, any new custom filters will appear as operators for use in the Transform toolbar.
Creating a Custom Filter

1. Choose custom in the choice box.
2. Enter a value for the center box. This is the brightness multiplication value for the target pixel. Most values used are quite small, almost always less than 20.
3. The boxes surrounding the center box represent pixels surrounding the target pixel. In each box, enter the convolution multiplication value to use for that pixel. For example, to multiply the brightness value by 3 of the pixel immediately to the left of the target pixel, place the number 3 in the box immediately to the left of the center box.
4. Not all boxes need to be filled. Boxes left empty will cause the pixels they represent not to be used for the convolution calculation.
5. Enter the value for scale to be used. Convolution works by multiplying the brightness values of pixels used by the number in their box and then adding up the result. This result is divided by the scale number. Larger scale numbers result in less overall brightness.
6. Enter the value of offset, if any, to be used. The offset number is added to the brightness values computed as a result of the convolution divided by the scale. Positive numbers for offset increase overall brightness. Negative numbers decrease overall brightness.

Examples

The following examples all apply different filters to the image of Tokyo airport above, shot from the Landsat 7 satellite.

Difference East

The Difference East filter subtracts values from the West and adds them to the East. This increases the brightness values where an edge difference exists to the East.
The **Low Pass 1** filter ends up eliminating changes that are only a pixel wide, and so blurs the image. Note how the convolution values surrounding the central value are all the same as the central value. Three different low pass filters provide differing values for the matrix.

"Low frequency" details are those where there is slow change from pixel to pixel. "High frequency" details are those where there is rapid change from pixel to pixel. The "low pass" filters are so named because they allow low frequency changes to pass but block high frequency changes.
The **High Pass 1** filter allows high frequency details to pass and blocks low frequency details.

Note how the image of Tokyo airport has had small changes from pixel to pixel emphasized by the high pass filter. Three different high pass filter presets provide differing matrix values.

**Edge Pixel Effects**

Because matrix filter effects work by convolution using values that surround a central pixel they raise the question: what to do with pixels on the very edge of the image, where there are no adjacent pixels on one or more sides to participate in the matrix? There are two approaches used by raster image editing packages. One approach is to simply leave pixels on the edge unchanged.

As may be seen above illustration, a zoomed in view of the result of the low pass filter, this is the Manifold method and is also the approach used by most professional image editing packages.
Another possible approach is to apply the matrix effect to edge pixels as if they were surrounded on the "empty" side by black pixels. Manifold does not take this approach because it results in inaccurate effects with edge pixels. If desired, an image can always have a black or other color border drawn about it if such an "approximate" effect is desired for edge pixels. In this way, the Manifold approach allows users to choose how they wish edge pixels to be treated.

5 x 5 Matrices

All preset convolution matrices in Manifold are 3 x 3 matrices: only the central value and the immediately adjacent pixel value boxes have values in them. All other values are zero. Surprisingly, most classic image manipulation effects may be achieved with 3 x 3 matrices.

If desired, we may create filter effects that employ 5 x 5 matrices by filling in the additional boxes with values. This will require longer computation time, but allows finer control over the convolution computation. Filtering with a 5 x 5 matrix that contains zero for boundary values will automatically switch into 3 x 3 mode for better performance (since a 5 x 5 matrix with zeros in the outermost boxes is equivalent to the 3 x 3 matrix that is surrounded by the zeroes).

Continued Education

The above three examples are just three of many filters provided with Manifold. Review of the other filters in the preset list box combined with experimentation using new, custom filters is a good way to learn how various convolution matrix-based filters work.

In addition, there are news groups and lists on Internet in which image manipulation effects are discussed. Participation in such groups and lists together with creative use of a good search engine will provide many sources of information on using and creating convolution matrix filters.
**Image - Fluoresce**

*Fluoresce* is a fun effect intended for artistic usage with RGB or RGBA images. It alters the lightness of colors in an asymmetric way so that some color ranges will appear unusually bright. A saturation control is provided since the next step is often adjustment of saturation.

- **Saturation**: Increase or decrease color saturation. Fully-saturated colors are very bright, while low saturation are grayish.
- **Lightness**: Increase or decrease lightness using differential fluorescence effect.
- **Preview**: Check to see effect in action.

Effects will vary depending on the image. The illustration above shows the sample *bronze* image with lightness lowered and saturation increased.

For artistic purposes, this effect works best with images where there is a great difference between very bright areas and other areas, such as the image above.
The image on the left shows the effect with lightness decreased for a fluorescent effect and the image on the right with lightness increased for a "blueprint" look.
**Image - Noise**

This command adds the desired amount of noise to RGB or RGBA images in the form of random color or monochrome color changes per pixel. Such changes will be often be perceived as adding small line segments, since human perception will attempt to "organize" random noise into distinguishable features.

- **Amount**  
  Move the slider to increase or decrease the amount of noise. Alternately, enter a value from 0 to 255 directly into the value box.

- **Monochromatic**  
  Check to add monochromatic noise only. This changes the lightness values of pixels without shifting their hues.

- **Preview**  
  Check to see the effect previewed in the image.

The image at left shows color noise, while the image at right has the same amount of monochromatic noise.

This command is also available on the Transform Toolbar for images as the **Add Noise** and **Add Noise(Mono)** transform operators. However, when used from the transform toolbar the effect is immediate with no preview possible.
Image - Gamma
Gamma is a measure of mid-tone color brightness. Changing gamma will result in a perceived image difference that is similar to changing the brightness. Used with RGB or RGBA images.

**Gamma**  
Increase or decrease gamma. Increasing gamma will increase overall brightness with a slightly greater effect on midtones.

**Preview**  
Check to see effect in action.

This command is also available on the Transform Toolbar for images as the **Gamma** transform operator. However, when used from the transform toolbar the effect is immediate with no preview possible. In the transform toolbar version the parameter specified is used as the Gamma level.
**Image - Gaussian Blur**

A Gaussian probability curve is the classic bell-shaped curve with a higher probability in the center of the curve and lower probabilities to either side. A Gaussian blur effect takes each pixel in an RGB or RGBA image and mixes it with adjacent pixels with Gaussian probability so that the pixel has great effect near its original location and less effect (in a bell curve shaped way) farther away from its original location. If a selection is present, the Gaussian blur will ignore any unselected pixels. Gaussian blur will also ignore invisible pixels.

- **Amount**  Number of pixels to blur to either side.
- **Preview**  Check to see effect in action.

This command is also available on the Transform Toolbar for images as the **Gaussian Blur** transform operator. However, when used from the transform toolbar the effect is immediate with no preview possible. In the transform toolbar version the parameter specified is used as the amount of pixels to blur.

**Example: Adding a Drop Shadow**

Gaussian blurs are most often used to create "drop shadow" effects.

Suppose we have an image layer that contains just the monument clipped out of our standard **bronze** sample image. We would like to add a drop shadow.

We begin by making a copy of the layer, selecting the monument only and then using Hue / Saturation to turn the **Lightness** all the way down. This makes the monument all black.
We next apply the Gaussian Blur effect, using an **Amount** setting of 5. The farther the "shadow" is supposed to be from the image, the greater the **Amount** should be for a credible effect.

We then position the initial layer over the drop shadow layer and then move either the shadow or the upper image so that the shadow is slightly offset. Drop shadows are often used to create the "cardboard cut out" effect seen at the bottom of the monument. If we wish to avoid this effect we can feather the bottom of the monument by applying partial transparency to the bottom with a partial eraser (assuming the image is an RGBA image in which partial pixel transparency is possible).

**Example: Adding a Halo**

Gaussian blurs are also used to create "halos" that set off images. There are two ways of doing so. The easiest way is to take the black blurred shadow and to colorize it and to then use the colorized "shadow" as a halo.

Here is a slightly longer version that shows the steps more explicitly.
If we take the black and white image created by reducing lightness, we can invert it to get a white image (shown here on a black background, since otherwise the white image would be invisible against a white background).

Next we use Colorize to convert the image to bright yellow. We did this by turning the Blue slider all the way down and the Red and Green sliders all the way up in the Colorize dialog.

We can now apply the Gaussian blur to blur the yellow image.
Here is the completed composition. The original image is in a layer above the yellow image, which has been offset slightly upward and to the left. We have also used three other effects:

- The saturation in the bronze monument has been increased to provide more vivid colors.
- The bronze monument is an RGBA image. We've used the eraser to partially erase the bottom part of the image to feather it into the background by virtue of increasing transparency.
- So the yellow halo does not show through the partially transparent monument image, we have erased most of the bottom half of the yellow halo to an even greater degree than the monument.

**Drop Shadows in Maps**

Drop shadows underneath image layers can add striking depth to maps.

The illustration above shows four image layers above several drawing layers in a map. The Manifold logo and text are both in separate layers and their shadows underneath are drop shadows in RGBA image layers created using Gaussian Blur. Both drop shadow layers have had transparency increased so their shadows are not so obvious. It is especially important when adding drop shadows to text to use a light touch. Increase transparency so there is barely a hint of shadow for a better effect.

Use drop shadows with logos, with text in image layers and to provide a "drop shadow" for photographs or other images that may be assembled "on top" of the map.

To create the image above, we zoomed into the map, made a screen shot, pasted the screen shot into a new image and then cropped the image down to the size of the detailed view desired. This image went into a layer in the map. We made a copy of the layer, turned the lightness down to zero to create a black square that was
exactly the same size as the cropped detail and then used Gaussian Blur on this black square to create a drop shadow. Finally, we offset the drop shadow and added the A and B labels.

See the Cookie Cutter a Large Image with Transfer Selection example topic for a detailed description of how to create a drop shadow in a geographic map.

**When a Selection is Present**

If a selection is present the Gaussian blur command will operate only on selected pixels.

For example, in the image above a circular region was selected and Gaussian blur applied. Only those pixels in the selected region were blurred.
Image - Invert

The Invert command is used to invert color values in pixels in grayscale, RGB or RGBA images. The Invert command is available in two forms: Invert and Invert At. Invert simply switches 0 to 255 and 255 to 0 and likewise switches mirror-fashion all values in between. Invert At accepts a parameter value that will stay unchanged and maps all values less than the parameter value to mirrored values greater than the parameter value and vice versa.

- **Center** Center point of the inversion (by default 128).
- **Preview** Check to see effect in action.

This command is also available on the Transform Toolbar for images as the Invert and Invert At transform operators. However, when used from the transform toolbar the effect is immediate with no preview possible. The parameter for Invert At is used for the center point of the inversion.

In a simple black and white image, using 0 for black and 255 for white, a near-black pixel value of 5 will be converted to 250, or near-white. The result is a photographic negative image.

For RGB color images or any other image with more than one channel this process is applied to each channel. If a pixel has a high blue value and low red and green values (like the blue sky in our sample image), the blue value will end up low and the red and green values will be high to result in a yellow tone.

The Invert At command includes a slider bar that allows setting the center point of the inversion process. Leave it at 128 to achieve a normal, symmetric inversion as discussed above. Increasing the center point increases the brightness of the inverted image, while decreasing the center point reduces the brightness of the inverted image.

The above example shows default inversion with a standard center point of 128 as well as brightened and darkened inversions at center points of 192 and 64 respectively. The center point control is provided because negatives are often unexpectedly light or dark and require immediate adjustment to suit one’s taste. Using the Invert command is effectively equivalent to using an Invert At command with a center point of 128.

**Comments**

The difference between the two commands is subtler than simply providing a center control for Invert At. The difference between the two is that:

- Invert always maps 0 to 255 and 255 to 0 (and, of course, inverts all other values in the middle), while
- Invert At with a center of 127 maps 0 to 254 (not 255) and 255 to 0, and
- Invert At with a center of 128 maps 0 to 255 and 255 to 1 (not 0).

The above may seem like splitting hairs, but when using Invert with surfaces the difference between 0 and 1 and 254 and 255 may be very significant.
Image - Motion Blur

**Motion Blur** provides a streaked blur effect as if the RGB or RGBA image were dragged along the direction indicated. In cartography, the main purpose of **Motion Blur** is as a directional "filter." It also may be used to create artistic effects to give a sense of motion to images.

- **Direction**  A number from zero to 180, showing the direction in degrees around a circle to use for the blur.
- **Amount**  Number of pixels to blur.
- **Preview**  Check to see effect in action.

This command is also available on the Transform Toolbar for images as the **Motion Blur** transform operators:

- **Motion Blur Diagonal 1**  Blurs the image as if moved diagonally on the upper left / lower right axis by the specified number of pixels.
- **Motion Blur Diagonal 2**  Blurs the image as if moved diagonally on the upper right / lower left axis by the specified number of pixels.
- **Motion Blur Horizontal**  Blurs the image as if moved horizontally by the specified number of pixels.
- **Motion Blur Vertical**  Blurs the image as if moved vertically by the specified number of pixels.

However, when used from the transform toolbar the effect is immediate with no preview possible. In the transform toolbar version the parameter specified is used as the amount of pixels to blur.

**Uses**

The obvious artistic usage of **Motion Blur** is to annoy people by making them think their eyes are out of focus:

A more responsible use of **Motion Blur** is to create a blurred version of the image that may then be placed underneath the unaltered image in a map layer. If we take the image shown above and place a cutout of the unblurred monument on top we can give the impression that the monument is moving.
To create this example we made a copy of the bronze image and selected all pixels except the monument. We then deleted those pixels to leave just the monument. This appears in a layer above the blurred image.

**A Cartographic Example**

As much fun as it is to ruin people's eyesight with blur effects, *Motion Blur* does in fact have great utility in vectorization. We use it to extract lines that run in particular directions.

Suppose we have a raster image that contains horizontal and vertical lines as well as many other lines. We want to extract just the horizontal lines.

Applying *Motion Blur* twice with maximum *Amount* setting and a *Direction* of 90 in both cases blurs everything but the horizontal lines. These don't appear to change because their pixels are moved onto each other. Everything else is blurred into various shades of gray.
We can now use Threshold to extract only those lines by setting the threshold range so that everything except the lines is made white.

Had we set the **Direction** to any other value we could have extracted features oriented in any other direction. Since raster images are rarely very accurate it is rare that we have the opportunity to use **Motion Blur** in this way. However, when we can do so the effect is like magic.
Image - Posterize
Converts the RGB or RGBA image to a limited number of colors for a "poster" effect.

Level
Number of color levels allowed in each channel. Very low values (2, 3, or 4) have the most visible effect.

Preview
Check to see effect in action.

This command is also available on the Transform Toolbar for images as the Posterize transform operator. However, when used from the transform toolbar the effect is immediate with no preview possible. In the transform toolbar version the parameter specified is used as the level.

Posterize is best used with small numbers of levels. It divides each channel into intervals of the specified number of levels and then builds colors within those intervals. For three, RGB channels the number of colors used in the image is the cube of the levels specified. For example, specifying 2 levels results in images with eight possible colors (2 R levels x 2 G levels x 2 B levels = 8 possible colors). Therefore, levels much above 6 produce little change.

For good "poster" effects, it’s often advantageous to follow posterize with a Brightness / Contrast adjustment to increase the contrast.
Image - Quantize

Quantize converts colors in RGB, RGBA or grayscale images to the specified number of colors using the desired method.

<table>
<thead>
<tr>
<th>Colors</th>
<th>Number of different colors to be used in the image.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preview</td>
<td>Check to see effect in action.</td>
</tr>
</tbody>
</table>

For example, using 8 colors with Quantize will transform the sample bronze image using only eight colors to render the image. The Quantize transform will choose and place eight colors to approximate the visual effect of the original colors.

Compare the results of Quantize with 6 colors in the image at above left with the image above right showing the Posterize command applied with a Level of 2, which also results in six colors as well. However, because Posterize uses free selection of color values within intervals it has more different colors. In contrast, Quantize results in an image drawn mostly in greens since these are the predominant tones in this image for visual effect.

A Cartographic Example

Paper maps are often created with the intention of using a limited range of standard colors. The printing process, however, results in small variations in color so that the actual printed map consists of thousands or even millions of colors that fall into several similar classes. When we scan a paper map (perhaps with the intention of using Manifold’s tracing tools to create a drawing) the scanner will create an image that uses a very large number of colors in what appears to our eyes to be a region of identical color.

Manifold’s selection tools allow us to set tolerances so that pixels of similar color can be selected easily. Sometimes, however, we would like to force colors in the image into a more limited number of colors. To do so we can use the Quantize command.
We begin with a scanned RGB image that contains many colors.

Using **Quantize** we can reduce the number of colors to only 6. This forces all the different "violet" colors used in some features to consist of only a single violet color.

We can then select that color using **SHIFT-select touch**, choose **Edit - Select Inverse** and then press **Delete** to delete all colors except the desired violet color.
The image has the **background** layer turned on so that everything except violet looks white; however, we have just created an image where everything that is not violet is a transparent pixel. To clean up the image we can use **Simplify** to eliminate small groups of pixels or single pixels. If desired, we could apply the above procedure to the red colored major roads to extract only those.

We could then combine the two images in a map so that one layer has the violet pixels and the other layer has the red pixels. This map would be a very useful map for tracing major roads shown in the scanned image. (The image above uses slightly different **Simplify** settings for the violet layer than in the previous images).

**Technical Note**

The *Manifold Quantize* transform uses the Wu algorithm to quantize colors with a modification to improve color matching by slightly preferring (weighting) G channel values over R channel values over B channel values. The weighting curves match the sensitivity of the human eye to color frequencies and thus provide a more “natural” look.
Image - Relief
Alters image to provide highlighting as though the RGB or RGBA image contains 3D objects illuminated by light from the side. Choose the direction of apparent lighting and then the degree of 3D relief desired.

This command uses the intensity of pixels as an "elevation" value if no other elevation data source is present. For images created from terrains, Relief will use the actual elevation data from the terrain to compute highlights and shading.

**Direction**  Direction toward which light is shining. "South" means it is shining towards the South.

**Highlights**  Apparent elevation scale to compute brighter, "lighted" points.

**Shadows**  Apparent elevation scale to compute darker, "shaded" points.

**Preview**  Check to see effect in action.

A South-East direction of relief was applied in the example above, so that light is apparently shining from the upper left down towards the lower right.

Moderate amounts of relief will provide useful effects when applied to many aerial photos, especially if the Direction chosen is aligned to the direction of light in the photo. The above example applied a North (upward pointing) direction of light to the image on the right. Since the image already contained visible shadows, a greater degree of Highlights was used than Shadows.
Relief may also be applied to non-geographic images to achieve a wide range of artistic embossing effects. Like all image commands, Relief applies to the entire image unless a selection is in effect, in which case it will apply to the selected pixels only. In the image above we selected a circular group of pixels and then applied Relief to achieve a circular region of embossed appearance.

When applied with restraint, Relief can give "hyper-real" effects to normal photographs, and can bring out hidden detail in geographic images, as seen below:
Image - Relink / Unlink

The Relink and Unlink commands are enabled when the focus is on a window for an image linked from an OGC WMS image server or from a Microsoft TerraServer image server.

- **Relink** - Used to reconnect a linked image to an image server. This command is also available within the linked image's context menu in the project pane. If a linked image is disconnected from an image server this command is used to reconnect it. For example, one might create a .map file that contains a linked image that is linked to an OGC WMS image server using a local URL. If the .map file is moved to another machine the link might not continue to work without adjusting the URL. In that case one can use the Relink command to reconnect the image in the project to the image server.

- **Unlink** - Unlink a linked image to create an image in the project. All data for the image is copied from the image server or cache if caching is enabled (the default) and embedded within the project as would be the case for an ordinary, unlinked image.

Image - Download and Image - Relink / Unlink Compared

An image linked from an image server may be converted to a local image within the project either by using the Image - Unlink command or the Image - Download command. The Unlink command converts the existing linked image to a local image at whatever resolution is currently in use. The Download command may be used to automatically fetch all image tiles at any available resolution to create a local image at that resolution.

Images linked from TerraServer may also be exported to image files. In that case, the exported image will be the same as that which would be obtained by unlinking the TerraServer image.

Note that exporting or unlinking an image linked from TerraServer will only use the data already downloaded from the server and will not attempt to download any more data. It will also only use the data for the most detailed image level. It is usually a good idea to use Image - Download to check how much data has already been downloaded for the most detailed image level prior to exporting or unlinking the image.

Another way to create a local image from a linked image is to open the linked image and then to use the keyboard F6 command or the Tools - Make Image command. The Make Image dialog allows us to create a local image in several different ways from the linked image without unlinking the linked image. For example, we can use this command to create an image of the displayed window or of the entire linked image at a desired number of pixels or scale.

**Very Important**

The Unlink command attempts to build a real image of the same pixel dimensions (width and height) of the linked image. Whatever tiles have been downloaded using the Download command will be used, and will be interpolated to cover those pixel extents. Keep that in mind if you want to unlink a huge linked image.

For example, suppose we create a linked image from a Manifold image server that covers the entire Earth. Images servers will now routinely offer to provide images at 0.3 meter resolution. If we link in such an image from an image server the linked image will be over 120 million x 120 million pixels in size. That's still easy to use within a Manifold project because zooming in or out or panning the image only shows a million or so pixels (1000 x 1000) in whatever viewport we have open, but if we unlink that image we are asking Manifold to create an image that is 120 million x 120 million pixels in size... that's over 50,000 terabytes in size at four bytes per pixel.

One might think that specifying a smaller number of tiles to download via Image - Download will do the trick to keep the data size down, since the Unlink command will use only the tiles downloaded. But suppose we download only those tiles representing, say, 100 kilometer x 100 kilometer resolution: the image we are asking Manifold to create is still 120 million x 120 million pixels in size, at 0.3 meters per pixel. What we are asking Manifold to do will be to take each of those limited number of 100 km x 100 km "pixels" and inflate them to cover the 300,000 x 300,000 pixels of 0.3 meters each within the larger blocks so that each such smaller pixel is the same color.

If what we want to do is to create an unlinked image with a smaller number of pixels, we should specify a lower resolution for the original linked image, choosing some value much greater than the 0.3 meter value for the image server we are using. For example, choosing, say, 5 kilometers results in a much smaller 7500 x 7500 pixel linked image. We can download all the tiles for that and Unlink and still have a relatively workable image.

See Also

Linked Images from OGC WMS Servers
Linked Images from TerraServer
Image - Simplify

Simplify is a very versatile effect that may be used for both pre-processing of RGB or RGBA images intended for vectorization as well as for purely artistic effects.

- **Level**: Increase to apply more effect (that is, greater simplification).
- **Neighbors**: The number of neighboring pixels to consider for each pixel.
- **Times**: The number of iterations.
- **Preview**: Check to see effect in action.

Simplify works by taking each pixel in the image and examining its neighboring pixels out to a given number. The pixel is then set to a color given by the weighted average of its neighbors. Increasing the **Times** value tells Manifold to repeat the process the given number of iterations. Repeating this process will eventually reach a point where for any given combination of **Level** and **Neighbors** settings there will be a **Times** setting after which no more changes occur.

When applied to the sample bronze image with a high **Level** will result in a pointillist effect.

Applied with a maximum **Level** the figure becomes abstract in its simplification.

A Cartographic Example
We can use the Threshold command to pull features of a given color out of a scanned raster map, such as the above section of a USGS DRG scanned raster map.

Simplify applied with a Neighbors value of 3 gets rid of the scattered dots in the image.
Image - Threshold

Threshold converts each pixel into black, white or unchanged depending on whether the original color value is within the threshold range. Threshold is a very important command that is often used to prepare scanned RGB or RGBA images for vectorization or use as guide layers in the creation of drawings. It can be used with raster data images to set off ranges of values that may then be used for subsequent analysis or as selection masks.

This command is also available on the Transform Toolbar for images as the Threshold, Threshold Black and Threshold White transform operators. However, when used from the transform toolbar the effect is immediate with no preview possible. The parameter in the transform toolbar is used to set the threshold between white and black.

- **[Histogram]** Shows distribution of intensity values. The taller the line, the more pixels have that intensity value.
- **[% number]** The percentage of pixels that are within the threshold range.
- **[Upper edit box]** Beginning of threshold range.
- **[Lower edit box]** End of threshold range.
- **[Left Mouse Click]** Clicking the left mouse button in the histogram sets the left side of the threshold range.
- **[Right Mouse Click]** Clicking the right mouse button in the histogram sets the right side of the threshold range.

**Modify** Choose standard, black or white threshold operation.

- **Both** - All pixels within range are forced white. All others are forced black.
- **Black** - All pixels within range are unchanged. All others are forced black.
- **White** - All pixels within range are forced white. All others are left unchanged.

**Preview** Check to see effect in action.

The two edit boxes set the lower and upper bounds of the threshold range.
Changing the lower box's value from 255 to a lower amount will constrict the threshold range. In the example above, all pixel values with intensity lower than 107 and all those with intensity 141 will be forced black. All those in between will be forced white. The white pixels will be 30% of the pixels in the image.

![Image showing threshold values and resulting image]

The result is an image where all intensities except those middle intensities are black.

**Black and White Options**

Choosing the **Black** option restricts the action of the **Threshold** command to only those pixels that are to be forced black. All other pixels are left unchanged. For example, using the threshold range above all pixels in intensity from 107 to 141 will be left unchanged. All other pixels will be forced black.

The **White** option restricts the action of the **Threshold** command to only those pixels that are to be forced white. In the above example, all pixels from 107 to 141 will be forced white. All other pixels will be left unchanged.

**Use with Color Images**

**Threshold** may be applied to RGB or other multi-channel images. When applied to color images, the histogram shows the intensity level of pixels. For per-channel control of threshold in color images, use the **Threshold Color** command.

We begin with the sample **schloss** image. “Schloss” is German for "castle" or "chateau." The image shows the famous fairy-tale castle, Schloss Neuschwannstein, built by King Ludwig in Bavaria.
The histogram for this image shows that over 40% of the pixels are clustered about a very high intensity. These are obviously the pixels that make up the sky in very high intensity, blue-white colors. We've moved the range controls so that for the most part only the sky pixels will be in range.

The **Both** setting forces all pixels in range to white and all those out of range to black. Since the range ends at 255, there are no pixels above the range. Only darker pixels are below.

Using the **Black** setting changes those pixels that are out of range to black and leaves those pixels in range unmodified. These are the sky pixels, which are left in their original light, blue-white colors.
Using the **White** setting changes those pixels in-range to white and leaves all other pixels unmodified. The above illustration looks similar to the original but on close examination will be seen to have only pure white pixels in the sky where colors used to be light, blue-white tones.

By careful selection of ranges, we can choose only a limited range of intensities to be forced white with all other pixels forced black. For example, the above range will leave only a limited range in white with all intensities above and below the range forced black.

The result is that we have picked out a particular range of intensities in the image. Note how both the very light sky (above the designated range in intensity) as well as the ground and trees (below the range) have both been forced black.

**A Cartographic Example**
Using a scanned image of a map as an example (a USGS DRG map) we can use threshold to suppress details except for the pink colors.

A bit of interactive experimentation shows that the intensity peak selected in the range in the histogram shown above is associated with the many pixels in the image in pink shades, all of which have about the same intensity.

Using **Black** as the modify operator, we force all pixels outside the narrow range to black, leaving only those within the intensity range as pink.
We can move the range to other "bumps" in the histogram to pick out other pixels by intensity ranges.

This range of intensities corresponds to the purple tones.

The next bump in the histogram corresponds to darker red tones.
In all the above examples, we force pixels outside the range to pure black color. This makes it easy to select those pixels using Select Touch and to delete them. Alternately, we could Invert the image to force them to white to make our job of vectorizing a set of lines that much easier. Note the profound simplification of the image in the last example. Note also there is nothing that stops us from using Threshold several times to pick out different "slices" of the image and to then combine them (perhaps after other operations such as Invert) to get exactly the image we want to use for vectorization. Very often the desired result will be reached by combining several commands.

In the above sequence it is tempting to imagine we picked out a particular color to be forced white; however, this is not the case. We have simply picked out a range of intensities. It is often the case in photographic images that particular colors or regions fall into similar intensity ranges so we can exploit this effect sometimes by using intensity as a proxy for color values. For threshold selection by true color values, use the Threshold Color command.

Another way of picking out particular color tones would be to use Select Touch to pick out all the pixels in dark red color. We could then choose Edit - Select Inverse to invert the selection and thus select all pixels except those in dark red color. Next, we could delete the selected pixels or use the Brightness / Contrast control to set the brightness of the selected pixels to zero, which would make them all black. There are often several different methods we can use in Manifold to achieve a desired result. Which method we choose depends on what is most convenient and most accurate at the time.

**Selections**

A reminder: Like all Manifold commands, Threshold will work only on the selected pixels if a selection is present.

In the sequence above, we've selected a "5" shaped region of pixels, applied Threshold and then turned off selection color so the effect can be seen. Cool!
Image - Threshold Color

For RGB or RGBa images, the Threshold Color command is similar to the Threshold command, but works on each individual RGB channel. Please see the Threshold command topic for details on the regular Threshold command.

Threshold Color is a very important command that is often used to prepare scanned images for vectorization or use as guide layers in the creation of drawings. It is also used to pick out colors and ranges of colors for graphics art editing and visual effects. It can be used with raster data images to pick out ranges of values to be used as selection masks.

Although it may be used in a simple interactive way to adjust images based on visual appearances using Preview, in sophisticated hands the Threshold Color dialog is a powerful instrument for choosing regions with specific color characteristics using Boolean combinations of RGB color values.

[Histogram] Shows distribution of intensity values. The taller the line, the more pixels have that intensity value.

[Upper edit box] Beginning of threshold range.

[Lower edit box] End of threshold range.

[Left Mouse Click] Clicking the left mouse button in the histogram sets the left side of the threshold range.

[Right Mouse Click] Clicking the right mouse button in the histogram sets the right side of the threshold range.

Modify Choose standard, black or white threshold operation.

Both - All pixels within range are forced white. All others are forced black.

Black - All pixels within range are unchanged. All others are forced black.

White - All pixels within range are forced white. All others are left unchanged.

Match Choose use of channels in deciding whether to make a pixel black or white (or left unchanged).

All Channels - If pixel is in range in all channels, make it white, otherwise make it black.

Any Channels - If pixel is in range in any channel, make it white, otherwise make it black.

Each Channel - Apply the modify operation to each channel independently. For each pixel, the R, G, and B values will set to 255 if it is in range for that particular channel and to 0 otherwise.

Preview Check to see effect in action.

Threshold Color provides three histogram threshold range panes, one each for the Red, Green and Blue channels. Each channel is considered independently, so the three histograms show the distribution of values for each channel independently of each other. The dialog opens with ranges set so that 50% of the pixels for each channel fall in range.
The histograms above show the RGB values for the sample bronze image. We apply various combinations of Modify and Match settings in the images below:

The default setting of Both forces all pixels within range to white and all pixels out of range to black. Using All channels means that a pixel’s values must be in range in all three R, G and B channels. Note that the brightest regions are forced white and the darkest regions are forced black.
Changing the modify setting to **White** forces pixels in range to white and makes no change (leaves colors unmodified) for this pixels out of range. Note how those pixels forced black in the first image in this sequence retain their original colors.

Using **Black** for the modify setting forces pixels out of range to black and leaves unchanged those pixels in range. Note how the clouds and lighter part of the sky are left unchanged while all the darker pixels are forced black.

Referring once more to the RGB histograms, we can use **Both** for the modify setting so that all pixels will end up either black or white, and now we can change the match setting to **Any Channel**.
If any pixel falls in range within any of its RGB ranges, it will be forced white. Because many pixels in the image are in range in at least one of the three channels, this means that very many pixels will be forced white.

The final case requires the greatest thought to understand. Using Each Channel means that each color range is considered entirely on its own. For each pixel, if the Red value is within range it is set to 255 (that is "forced white"). If it is not in range it is set to 0 ("forced black"). This process is repeated for the Green and Blue values.

Using Each Channel with Both set means that every pixel in the image will be either black, white, red, blue, green, cyan, magenta or yellow because these are all the possible combinations of pixels having all or nothing values in the Red, Green and Blue channels. For information on why this is so see the Images and Channels topic.

The bronze monument ends up yellow because pixels there tend to have lots of red and green (so both the R and G channels are forced full ON), without as much blue. 50% of the high blue pixels are mostly in the sky and clouds, so the default settings for ranges tend to push the B values in the monument to zero. If R and G are 255 and B is zero the result is yellow.

A Cartographic Example

When vectorizing scanned map images into vector drawings, we can often improve the performance of our tools by carefully preparing the scanned image with Manifold image tools before beginning vectorization. Sophisticated use of the Threshold Color command will help us do this. This example shows how Threshold Color can be used to pick out regions of different colors.
For a sample image we will use a USGS DRG format scanned raster map downloaded from the USGS web site. Let's begin by getting rid of all of the black pixels.

To do this, we set up the channel controls with values from 0 to 10 for R, G and B with Modify set to White and a Match for Any Channel. We are telling Manifold to change to white all pixels within the range. The only pixels with such low values in RGB are black or near-black pixels, so this changes all of them to white. We apply this effect so there will be no more black pixels.

To make all pixels white except the bright red ones, we retain the White and Any Channel settings in the Modify and Match controls and choose a range of low values for red and high values for greens and blues. This turns everything white except those pixels with midrange or high values for Red that also have low values for Green and Blue. The result would be to make everything white except the bright red lines. Instead of applying this effect, let's see what we have to do to get the purple lines.
These are more difficult because they include Red color in their pixels as well as Blue and Green. We begin by
setting all of the Red lines to white. Choosing any channel that has the above settings gets rid of bright red while
retaining pink and purple. We apply this effect.

If we now choose high brightness value ranges for all three channels we get rid of the very lightest tones no
matter what their color. The purple colors remain because they are not as bright as the light pink tones.

Notes

When looking at the three RGB histograms it is easy to make the mistake of thinking that a vertical line through
the three histograms will show the R, G and B value for pixels on that line; however, this is a wrong understanding
of the histograms. The histograms simply show the distribution of pixels by the total number of pixels at each R,
G and B level. The histograms do not mean that these are the same pixels at each level even though that might
often be the case. Certain color tones will often have similar values in R, G and B. For example, white clouds in
a sky will show peaks in all three histograms because many pixels in those clouds will simultaneously have high
values in all three channels.

To get the same pixel at each level, we need to average the R, G and B values into a combined intensity, which is
how the regular Threshold command works. In many instances we can use the intensity as a proxy to pick out
specific color shades. See the Threshold command topic for an example.
Image - Tile
Divides an RGB or RGBA image into tiles of given X and Y extent and then averages pixel values within the tiles to derive a single color that is then applied to the entire tile. This is used for aggregating and interpolating raster data images into larger tiles and for creating "pixelated" artistic effects.

This command is also available on the Transform Toolbar for images as the Tile transform operator. However, when used from the transform toolbar the effect is immediate with no preview possible. The parameter for Tile in the transform toolbar is used for both the X and Y pixel tile size.

- **X Tile**: Size of tiles in pixels in horizontal direction.
- **Y Tile**: Size of tiles in pixels in vertical direction.
- **Use same X and Y**: When checked, any change in X or Y the other value as well so that X and Y values are always the same.
- **Preserve Colors**: If checked, will use a median computation to derive the color to be used for the tile from only those colors that already exist in the image.
- **Preview**: Check to see effect in action.

If we begin with the sample bronze image, using a value of 10 for both X and Y converts the image into tiles that are 10 pixels square where each tile is a single, solid color.

Using different values for X and Y results in rectangular tiles. The above image was created with X value of 16 and Y value of 4. The resulting tiles are sixteen pixels wide and four pixels high.
Using an X of 1 pixel and a Y of 16 results in an interesting artistic effect.

Preserve Colors Checkbox

The Preserve Colors checkbox controls how Tile determines what color to use for the tile. If this box is not checked, Tile will average the RGB values for all pixels within the tile to determine what RGB values represent an arithmetic average. This may pick out a new color that is not yet used within the image. When the Preserve Colors checkbox is checked, Tile will use a median computation to find the color already used within the image that is a best fit to the desired, interpolated color to use for the tile.

We use the Preserve Colors checkbox if this image was converted from a limited palette image and we do not want any new colors introduced into the image.
Image - Resize
Change the horizontal and/or vertical size of an RGB or RGBA image using bicubic, bilinear or nearest neighbor interpolation.

<table>
<thead>
<tr>
<th>Method</th>
<th>Bicubic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Width</td>
<td>186 pixels</td>
</tr>
<tr>
<td>Height</td>
<td>160 pixels</td>
</tr>
<tr>
<td>Constrain Proportions</td>
<td>✔</td>
</tr>
</tbody>
</table>

Method: Choose the interpolation method to be used to recompute new pixel values as pixels are added or removed to achieve the new size.

Width: Width of the resized image in pixels.

Height: Height of the resized image in pixels.

Constrain Proportions: Checking this box will cause any change in Width or Height to automatically update the other parameter so that the width to height proportion is kept the same.

The size readout at the bottom of the dialog gives the size of the image in kilobytes at the specified width and height.

The Method box chooses the mathematical technique used to interpolate the existing pixels into a greater or lesser number of pixels as the image changes size. Different methods will at times work better or worse depending on the image content. Choose from Bicubic, Bilinear and Nearest Neighbor. The default Bicubic method works best with most images.
The Convert To command converts images into different methods of handling color: RGBA, RGB, Grayscale and Palette images. Convert To changes the image data permanently. The Convert To dialog will also report the estimated size of the resulting image. The Convert To command may also be used to convert linked images into local images within the project.

**Image - Convert To**

Target Desired type of image: Compressed, Grayscale, RGB, RGBA or Palette.

Palette Choose one of the preset palettes for a specific usage, when converting to a Palette image.

Metric Method for computing proximity of colors to each other when judging if colors are close to each other in the color cube. Displayed when converting to palette images. Use Euclidean: the other settings are rarely used metrics provided for expert use in special situations.

Colors The number of colors to use to create the image. Displayed when converting to palette images.

Dither Check to use dithering. Displayed when converting to palette images.

Ratio Compression ratio. Displayed when converting to compressed images. Larger Ratio numbers will result in greater compression at the cost of lesser image quality. A ratio of 10 will create a compressed image that is one-tenth the size of the original image.

Save result as new component Instead of converting the same image component, places the converted image into a new component.

The Target box will contain only those choices other than the method currently employed for the image. RGB images, for example, will only have Palette and Grayscale appear as choices.

See the Dither topic for information on dithering.

**Palettes**

Adaptive Sample colors appearing most frequently in the image and use these to choose a best fit to the given number of colors from the entire RGB space.

Linear Choose colors that are evenly spaced in the RGB cube space for the given number of colors.

Photoshop Use the 90-color Photoshop swatches palette. Handy for many restricted-color graphics arts images.

System Use the 256-color Microsoft Windows system palette.

System (Mac) Use the 256-color Apple Mac system palette.

VisiBone 2 Use the VisiBone web designers palette.

Web Use standard 216-color palette included by default in all web browsers.

Palettes used by other applications and systems are provides so that images destined for those systems can be created using an exact match of palettes.

Grayscale and palette images may be converted into RGB images without any loss of color values. Thus, one may take a grayscale image, convert it into RGB, and then convert it back into grayscale without any loss of color data.
The reverse is not true: when converting RGB images into grayscale or palette images there will be a permanent loss of color information as the RGB color information is interpolated down into a much smaller set of tones that are available in grayscale or palette images.

RGBA are RGB images with an extra channel that allows setting a percentage transparency for each pixel. This is called an "alpha" channel. RGBA images are used when many smaller images are stacked together in a map to achieve photocomposition effects as is done with professional image editors such as Adobe PhotoShop. See the Layers topic for an example.

Compressed Images

Compressed images are decompressed when converted into other image types. Since compressed images are often very, very large be careful to consider the resulting size of image when converting into other image types. Keep in mind that compressed images also use a sophisticated display algorithm to more rapidly display the compressed data. When decompressed and converted into other image types the speed of redisplay available with a compressed image will no longer be available.

When converting images into Compressed form, although the visual appearance of the image will be almost exactly the same (or even may appear to be enhanced) the actual information content of the image will be reduced. The high-speed compression algorithms used are normally lossy algorithms based upon wavelet compression or similar compression strategies. Although they preserve or enhance visual appearance, the specific pixel values will change.

Consider the RGB image seen at high zoom level above. We have zoomed in so that individual pixels are visible. The scene illustrated is the Northern part of Baja California in Mexico and the Sea of Cortez.

After conversion to a compressed image, the scene at the same zoom level appears above. Note that individual pixels are no longer discernable because the compression technology used synthesizes a scene using full screen resolution to utilize however many pixels are available in the computer monitor's display.

Although the scene looks very realistic (considering the high zoom into what was originally a low-resolution image), it is very important for scientific purposes to understand that the information content in the compressed
image is less than that of the original. The compressed image only looks good visually because a clever decompression-on-the-fly algorithm was used to synthesize a visually appealing image from a smaller number of bytes.

When converting to other image types a compressed image can expand into a dramatically larger image in terms of storage space required. Pay careful attention to the estimated size reported by the Convert To dialog for the resulting image to avoid creating unexpectedly large images.

**Tech Tip**

Read only images (for example, from an Enterprise server if Enterprise Edition is being used) can be converted to other types of images if the **Save result as new component** box is checked.

Attempting to convert an image to a compressed format using **Convert To** will display any error messages returned by the ECW / JPEG2000 compression code should the conversion fail.

**Table Menu**

**Table - Add Active Column**

See the Active Columns topic. The **Edit Active Column** dialog is the same dialog that is titled differently and used when editing an existing Active Column.
Table - Add Rank Column

See the Rank Columns, Decision Support System and Add Rank Column Dialog topics.
Table - Address - Geocode

The Geocode command takes an address in standard form, finds it in the Manifold Geocoding Data product and then produces latitude and longitude values for the location of the address. Addresses to be geocoded must include the four standard Address, City, State and Zip fields.

Records that cannot be located within the error level specified by the Fail on parameter will be reported in an Unmatched Records dialog to allow editing and selection of proposed matches the system determines may be the correct locations. If a particular address cannot be located on a street, the system will report "Building not found" and will offer a match to the address. Choosing that match will choose a location for the address that is at the midpoint of the street segment for the street of that name.

Records that cannot be geocoded at all, even after manual intervention, will have zero values for latitude and longitude. When the table is copied and pasted as a drawing these records will be ignored if the Skip zero latitude / longitude records option is checked (the default setting) in the Paste As Drawing dialog.

If an address cannot be found in the available street address ranges for a specific street the command will choose a point near the middle of the street segment.

Geocode Addresses Dialog Controls

- **generate ... using** Specify the columns to be used for Longitude and Latitude, Status and Match type fields, and which columns in the table should be used for the standardized Address, City, State, and Zip fields.
- **Longitude** Table column to be used for the longitude of geocoded record. Contains 0 if the record was not geocoded. Use [New Column] to automatically create a new column called Longitude.
- **Latitude** Table column to be used for the latitude of geocoded record. Contains 0 if the record was not geocoded. Use [New Column] to automatically create a new column called Latitude.
- **Status** If desired, add a column that reports the status of a particular record after processing. [None] does not report status, [New Column] will add a new column to the table called Status.
- **Match type** If desired, add a column that reports the geocoding match type of a particular record after processing. [None] does not report match type, [New Column] will add a new column to the table called Match type.
- **Fail on** Criterion to be used to declare a particular record to be unmatched.
- **Offset Locations by** Offset the location for each geocoded record by the given number of units from the street line, to the left or right side of the street based on whether the address was found in the left or right range.
- **Skip completed records** Do not process records that already have values in the designated Address, City, State and Zip fields.

Fail on Options

- **any error** Does not process record unless it is a perfect match in all fields. Equivalent to setting an "unknown building" error.
- **unknown street name, possible misspelling** Matches to Zip and City and does not process record unless the street name is an exact match.
- **unknown street name,** Matches to Zip and City, and attempts to match to similar
no similar names  street names if an exact match to the street name cannot be found. Uses "Soundex" and similar algorithms to attempt to find the right street in case of a street name misspelling. This is the default setting.

unknown zip  Do not process record if the zip code cannot be found.

critical error  Stop processing only on problems with an incomplete geocoding database or on hardware failure.

Match type Values

When reporting the status of a record's processing in a Match Type field the following values are used, listed in order from most precise to least precise:

building  The zip code, street name and building number were used to identify a location at the building's number.

street  The zip code and acceptable street name were found but not the building's number so the location has been placed at the center of the bounding box for the street.

zip  Only a zip code was used so the location has been placed at the center of the bounding box for the zip code.

city  Only the city name was used so the location has been placed at the center of the bounding box for the zip code.

((empty string))  The record could not be geocoded and values of 0 have been written into both Latitude and Longitude.

Selections

Both the Standardize and the Geocode commands are auto-scoped: if a selection is present in the table they will operate only on the selected records.

See Also

Street Address Geocoding
Table - Address - Standardize
No Data
Table - Design

Launches design dialog to change the structure of a table. In tables, columns are fields. We use the words "field" and "column" interchangeably. The Design dialog is resizable.

This dialog may be used to alter the structure of linked tables as well as local tables. It is therefore a popular way of administering database tables even in databases outside of Manifold. To do so, use the Database Console to open the data source and link a desired table into the Manifold project. Next, open the linked table and use the Design dialog to alter the structure.

- **New Column** - Add a new column (field). By default, columns are added as 32 bit integers.
- **Delete** - Delete highlighted column.
- **Move to Top** - Move the highlighted column to the top of the column stack. This displays it as the leftmost column in a table.
- **Move Up** - Move the highlighted column up one position in the column stack. This displays it one column position to the left in a table.
- **Move Down** - Move the highlighted column down one position in the column stack. This displays it one column position to the right in a table.
- **Move to Bottom** - Move the highlighted column to the bottom of the column stack. This displays it as the rightmost column in a table.
- **Edit Values** - Reserved for future expansion.
- **View Default Values** - Show a column in the design pane for default values field.
- **View Types** - Show a column in the design pane for database types for each field.
- **View Transfer Rules** - Show transfer rules for each column. The transfer rules may be adjusted by double-clicking into a rule.
- **View Extended Properties** - Enabled for linked tables. Show extended properties in a "Unique" column. A typical extended property is to note which field is the primary key.

To add a new field to a table:

1. Open the table.
2. Choose Table - Design
3. Click on the Add Field button. A new field appears in the design pane.
4. Double click into the Name field and change the name as desired.
5. Click in the View Types button. Type and Size columns appear in the design pane.
6. Double click into the Type column for the new field and change the database type as desired for this field.
7. Double click into the Size column for the new field and specify the size as desired. Some defaults, such as a size of 1 for 8-bit integers, cannot be changed.
8. Press OK.

When adding a new field to a linked table, the Unique column will be enabled and a combo box will allow specifying the value of the Unique column to primary key, unique or blank. This column is made visible by pushing in the View Extended Properties button in the Design dialog's toolbar.
Default Values

Some database operations and record-creation operations benefit from knowing what a default value for a particular field is supposed to be.

To specify defaults for any field, first click the View Default Values button to show a Default Value column in the design pane. Double click into the Default Value cell for a field to specify a default value that should be used.

See Also

Database Console
Table - Relations
**Table - Relations**

Launches relations dialog to create and manage relations between two tables. **Relations** are connections between two tables that allow one table to show columns from another table. A relation between tables uses a key field with unique values common to both tables to connect the two tables. This allows data that appears in one table to also appear in another table without having to physically duplicate the data. The **Relations** dialog is resizable.

**To form a relation between two tables**

1. Open the table that is to display the additional fields and click on **Table - Relations**.
2. Click on the **New Relation** button. This launches the **Add Relation** dialog.
3. In the **Add Relation** dialog, choose another table from the list box.
4. In the **Add Relation** dialog, click on one field for each table that will be used to match records and press **OK**.
5. Back in the **Relations** dialog, check the desired columns from the other table. Press **OK**.

Columns that are "borrowed" from another table will appear in the table with yellow background color to indicate they are imported. They may be used like any other column, for example, for sorting, filtering or within formulas or thematic formatting. Tables may have more than one relation with more than one other table.

Columns that are included from other tables may not be used as a key field to form a new relation. Columns can only be linked through one relation. They cannot be passed on in turn through yet another relation. For example, if table A has a relation with table B so that columns B1 and B2 appear in table A, table C cannot form a relation with table A and "borrow" columns B1 and B2 from table A. To include columns B1 and B2 in table C, table C must form a relation directly with table B.

**Relations Dialog Commands**

- **New Relation** - Add a new relation.
- **Delete** - Delete highlighted relation.
- **Include All** - Show all columns from the related table in this table. Check all checkboxes.
- **Include None** - Do not show any columns from the related table in this table. Uncheck all checkboxes.
- **Include Inverse** - Uncheck all checked columns and check all unchecked columns. A fast way to show all but one column: click **Include None**, check the one column not desired and then click **Include Inverse**.
- **Properties** - Edit the highlighted relation.

**Example**

Relations are used to add columns to tables from other tables.
Suppose we have a Customers table with a CustomerID field and a ContactName field.

Suppose also we have an Orders table with OrderID, CustomerID and an OrderDate field.

We would like to create a relation that shows in the Customers table the OrderID and OrderDate columns so that for each customer we can see all orders for that customer as well as the date of order. To do this, we open the Customers table and choose Table - Relations.

In the Table Relations dialog we click the New Relation button to add a relation. This opens the Add Relation dialog.

In this dialog we choose the Orders table in the list box and then highlight CustomerID as the linking field in both the Customers table (the left pane) as well as in the Orders table (the right pane, under the list box showing the Orders table). Note that we could have used any two fields that contain matching values even if they are named differently. However, as a matter of sensible database organization it makes sense to give the same name to the field used to save customer identification in all tables. We press OK.
The result back in the Relations pane is that we have a new relation listed as CustomerID: Orders.CustomerID. This is a shorthand way of saying that this relation is determined by matching values in the CustomerID field in this table to the CustomerID field in the Orders table.

We check the fields (columns) we would like to include from the Orders table and press OK.

The result in the Customers table window is that two new columns appear, the contents of which are taken from the Orders table. These contents are shown in yellow, which is a generic background color used to indicate values that are computed or otherwise derived.

Notes

A table may have more than one relation. For example Table A might include two columns, B1 and B2, by way of a relation with Table B and Table A might also simultaneously include one more column, C1, by way of a relation with Table C.

Relations can exist between any tables that exist in the project. In particular, a drawing’s table can have a relation with another table.

Relations can exist between tables that are imported into a project and tables that are linked into a project.
When including fields from tables that are linked into a project, keep in mind that tables linked into a project may be provided by files or OLE DB providers that might be participating in multi-user sessions with other programs. That's usually the objective of including such external tables, of course, but one should not be surprised if data in columns that are included from external tables appears to change without user intervention.
Table - Relink / Unlink

The Relink and Unlink commands are enabled when a linked table has the focus.

- **Relink** - Used to reconnect a linked table to a data source. This command is also available within the table's context menu in the project pane. If a linked table is disconnected from a data source this command is used to reconnect it. For example, one might create a .map file that contains a linked table that is linked to an Access .mdb file. If the .map file and the .mdb file are moved to another machine the link might not continue to work. In that case one can use the Relink command to reconnect the table in the project to the .mdb file.

- **Unlink** - Unlink a linked table to create a table in the project. All data for the table is copied from the data source and embedded within the project as would be the case for an ordinary, unlinked table.

See Also

Importing and Linking Tables
**Table - Match**

The Table - Match command provides spatial geocoding within Manifold System. Match can be used to geocode tables by matching locations to drawings of postal codes, provinces, regions, counties or other geographic entities. Such drawings/maps are much more frequently available in international settings than are detailed data sets necessary for address geocoding. In addition, Match can be used to spatially geocode tables of all types using drawings as a guide. For example a table of well drilling information can be geocoded using a drawing of well locations as a guide.

**Controls**

- **Match** - Choose a drawing to which the table should be spatially matched.
- **Move to Top** - Move the highlighted field to the top of the field list. This gives it the highest priority in matching.
- **Move Up** - Move the highlighted field up one position in the field list.
- **Move Down** - Move the highlighted field down one position in the field list.
- **Move to Bottom** - Move the highlighted field to the bottom of the field list. This gives it the lowest priority in matching.
- **Use Case** - Consider upper or lower case when matching text values in fields. When pressed, "nAme" will not match "Name" or "name".
- **Use Side Whitespace** - Consider space characters occurring before or after text strings. If pressed, " 89701" will not match "89701" (with two spaces before the 8).
- **Use Interior Whitespace** - Consider space characters occurring within text strings. If pressed, "8 9 7 0 1" will not match "89 701".
- **Match All** - If pressed, requires a match to all fields.

**Column / Match To Pane** - Choose fields that must be matched to guide a spatial match.

- **X** - Field name to save longitude / x data. Suggested name: "Longitude".
- **Y** - Field name to save latitude / y data. Suggested name: "Latitude".

**Latitude / Longitude coordinates** - Write location in decimal degrees latitude and longitude.

**Match** is easy to use. If we have a database table that has one field in common with a drawing, we can use the drawing to geocode the table.

To spatially geocode a table using Match:

1. Open the table.
2. Choose Table - Match - Drawing
3. In the Match Drawing dialog choose the name of the drawing to serve as a spatial guide.
4. Choose the fields in the table that are to be matched to fields in the drawing.
5. Press control options as desired for matching text values.
6. In the X and Y boxes specify a name to use for the longitude field and latitude field.
7. Verify the Latitude / Longitude coordinates box is checked and press OK.

Each record in the table will be compared to all objects in the drawing for the specified fields. If the record matches field values for an object in the drawing the latitude and longitude of that object’s centroid will be written into the records latitude and longitude fields.

If only one field is specified, the records in the table will be geocoded by matching objects against that single field. If two or more fields are specified, Manifold will try to match in order.

See Also

See the Spatial Geocoding with Match topic for an example.

Script Menu

The Script menu appears in the main menu whenever a script window has the focus. It provides a menu of commands used to develop and debug scripts.

- **Open Form**: Open the form associated with this script.
- **Open Table**: Open the table associated with this script.
- **Run**: Run the script
- **Run under Debugger**: Run the script using the debugger. Available only if the script is written in an ActiveX language that supports the Microsoft debugging interface and if a Manifold edition that includes the Manifold debugger is installed.
- **Pause**: Pause running the script.
- **Stop**: Stop running the script.
- **Step Into**: Step into a function.
- **Step Out**: Step out of a function.
- **Step Over**: Step over a function.
- **Language**: Choose a scripting language.
- **References**: Available whenever a script window using a .NET language is opened. Allows specification of external .NET modules to which the script may refer.
- **Compile to DLL**: Available whenever a script window using a .NET language is opened.

In addition, the script Properties context menu pops up when we right click on a script in the project pane and choose Properties. This menu allows us to check the Run in separate thread property. Running a script in a separate thread allows using advanced scripting techniques such as scripting the user interface, but might be slightly slower in some cases. By default, the option is turned off.

See Also

Debugger
Script - Compile to DLL
Enabled when a script window using a .NET language is opened. This command allows compilation of the .NET script into a DLL.

When a script using a .NET language is created in Manifold it will be automatically compiled when it is run. Using the Compile to DLL command allows us to compile the script into a DLL before it is run. A DLL compiled this way will be identical to a DLL compiled with Microsoft Visual Studio and will contain the code for .NET entities defined in the script component.

One might want to compile a .NET script into a DLL for several reasons:
- To share the code between scripts.
- To expose code originally written as a Manifold script to other programs.
- To reduce the time it takes to launch a script by eliminating the need to compile it before running.

To compile a .dll from a .NET language script:

1. Open the .NET language script.
2. Choose Script - Compile to DLL.
3. In the Save As dialog, browse to a folder location and provide a filename. Press Save.
4. Any errors during compilation will be logged in the Errors pane.

The Errors pane displays the description, module, line and line position of any errors that occur during the compilation phase of running a .NET script. Highlighting an error in the Errors pane and clicking Go To (or, simply double-clicking the error) will open the relevant script component and place the cursor at the location of the error.

See Also

View - Panes - Errors
Script - Language

Manifold supports programming in scripting languages installed in your computer. The Script - Language dialog allows choice of which scripting language is used. VBscript (Microsoft Visual Basic Scripting Edition) is the default choice. Any ActiveX or .NET scripting language installed in your Windows system may be used. However, only VBscript may be programmed using Manifold's drag and drop, forms based programming facilities. Other languages must be programmed (in the usual way) using text interfaces.

Choose a scripting language from the Engine dialog. ActiveX and .NET languages installed in your system will be available in this dialog. The language selected will thereafter be used for that script. By default, since Manifold requires installation of IE both VBscript and Jscript are always available. Free downloads from numerous sites provide various other languages as ActiveX scripting engines. For example, one can download ActivePython or ActivePerl from the www.activestate.com site.

The dialog reports the scripting language, the version and the path for ActiveX scripting engines. When a script is read-only (that is linked from an Enterprise server if Enterprise Edition is used), this dialog will still report the language used by the script even though the language may not be changed for a read-only script.

The project pane status bar will also report the language used by a script component when the script is highlighted in the project pane.

.NET Languages

.NET languages like C#, Jscript.NET and VB.NET may also be used within Manifold scripts. If a .NET language is used, the Script - References command will be enabled, allowing use of external .NET modules within a script.

See Also

Please see the Programming Manifold topic for a discussion of scripting languages in Manifold.

See the Scripts topic for convenient keyboard shortcuts when working with scripts.
**Script - References**

The **Script - References** dialog is available whenever a script window using a .NET language as the Script - Language is opened. The dialog allows specification of external .NET modules to which the script may refer.

For example, one can refer to `System.Windows.Forms.dll` and have a script display a form. The resulting script can then be shared on an Enterprise server (if Enterprise Edition is installed) and reused in other projects, not something that can be done with Forms components.

When a script is read-only (that is linked from an Enterprise server if Enterprise Edition is used), this dialog will report the references used by the script.

- **New** - Add a new module.
- **Delete** - Delete highlighted module.
- **Move to Top** - Move the highlighted module to the top of the module stack.
- **Move Up** - Move the highlighted module up one position in the module stack.
- **Move Down** - Move the highlighted module down one position in the module stack.
- **Move to Bottom** - Move the highlighted module to the bottom of the module stack.

When adding a new module, keep in mind that system modules such as `System.Data`, `System.Windows.Forms` or `System.Xml` are usually located in

```
C:\Windows\Microsoft.NET\Framework\vXXX
```

...where **XXX** is the version of the .NET framework installed.

**See Also**

**Script - Language**

**Surface Menu**

**Surface Menu**

The **Surface** menu appears in the main menu whenever a surface window has the focus or a surface layer in a map is the active layer. Commands listed below will appear either under the main **Surface** menu or under submenus such as **Adjust** or **Effects**.

- **Invert** - Invert pixel height value. Creates a topographic negative so that heights become low regions and vice versa.
- **Threshold** - Forces height pixels to maximum, minimum or unchanged based on a histogram.
- **Posterize** - Convert surface to a limited number of height values. Height values are chosen to be evenly spread throughout the total range of heights for the number of levels specified.
- **Filter** - Applies convolution filters, either custom filters or a choice from numerous presets.
- **Noise** - Add random height values.
<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Tile</strong></td>
<td>Convert surface into rectangular tiled regions of appropriate height.</td>
</tr>
<tr>
<td><strong>Quantize</strong></td>
<td>Convert surface to the specified number of height values using the most frequently occurring heights.</td>
</tr>
<tr>
<td><strong>Resize</strong></td>
<td>Change the number of pixels in the surface using desired interpolation method.</td>
</tr>
<tr>
<td><strong>Convert To</strong></td>
<td>Convert the numeric types used for height information in the surface.</td>
</tr>
<tr>
<td><strong>Contours</strong></td>
<td>Create contours using this surface.</td>
</tr>
</tbody>
</table>
**Surface - Invert**

Invert pixel height value. Creates a topographic negative so that heights become low regions and vice versa.

- **Center**  The percentile about which inversion should be calculated. The illustration above shows a 50 percent inversion, which evenly flips highs and lows.

- **Preview**  Check to see the modification in action.
Surface - Threshold
Forces height pixels to maximum, minimum or unchanged based on a histogram. All height values below the lower range or above the upper range will be set to the lowest value (or left unaltered). All height values within the range can be set to the highest value (or left unaltered).

The above threshold conversion was done modifying all the lower values ("black") below 500 to the value 500 and thus the lowest color in the palette.

All of the values above 500 were left unchanged.

---

Values
Choose the minimum and maximum values for the modification range. The modification range is shown in white (lighter color) in the histogram with values outside of the range shown in black (darker color).

[Histogram Display] Left click to set the lower range and right click to set the upper range. The numeric readout shows the percentage of height pixels falling within the modification range.

Modify
Choose Both to modify values within as well as outside of the specified range. Choose Black to modify only those values outside the range. Choose White to modify only those values inside the range.

Preview
Check to see the modification in action.

The Threshold command can be used to cut contours out of a surface within a desired range.
For example, the dialog above has been set up to modify all pixels outside the specified range.

The result shows a "flat" terrain of low height everywhere except the narrow band of elevations within the specified range that are left unmodified.
**Surface - Posterize**

Convert surface to a limited number of height values. Height values are chosen to be evenly spread throughout the total range of heights for the number of levels specified.

**Levels**  Number of height intervals into which the total height range of the surface should be divided.

**Preview**  Check to see the modification in action.

For a surface ranging from 0 to 100 meters choosing a **Level** of 10 will resample the surface into only ten different height values every ten meters. The screen shot above used 8 levels.
Surface - Filter

The Filter command exposes the internal functioning of the convolution filter operators in the Transform toolbar for surfaces and allows us to use custom commands using convolution filters. Most users will apply the commands in the Transform toolbar without ever examining their matrix values in the Filter command. However, for those users who are so inclined, the Filter command provides an "open door" to altering existing filters and creating new ones.

Many computational effects for images and surfaces are created through a mathematical process called convolution. When applied to surfaces, the heights of pixels are recomputed based on the values of surrounding pixels. The rules for how each pixel's height value should be computed are given by a matrix of numbers, where the central number gives the height of the pixel and the other numbers set forth how values from surrounding pixels should be multiplied into the convolution. The illustration above shows the effect of the Difference West filter.

Choosing different numbers for the convolution matrix results in a very wide range of visual effects such as sharpening, blur and so on. Manifold provides a 5 x 5 matrix, where each pixel may be adjusted based on reference to 24 surrounding pixels. This convolution matrix will be applied in turn to each pixel in the image.

[List Box] Choose a preset filter from a list. Choose Custom to create a new filter.

Apply Load the convolution matrix with the numbers for the filter currently shown in the preset box.
Save As… Save current convolution matrix as a preset filter.
Delete Delete previously-saved preset filter or disable factory preset.
Preview Check to see effect in action.

The list box at the top of the dialog allows choice of a filter from many preset filters. Because there are many presets and because at times very large surfaces may be manipulated in Manifold, choosing a preset does not apply that preset's convolution matrix numbers to the matrix until the Apply button is pressed. This makes it possible to browse the presets without applying their convolution matrix to the main surface regardless of whether the Preview box is checked or not.

The Filter dialog has a choice box at the top for preset filter effects.

Applying a Custom Filter

1. Choose a starting preset in the choice box.
2. Enter a value for the center box. This is the height multiplication value for the target pixel. Most values used are quite small, almost always less than 20.
3. The boxes surrounding the center box represent pixels surrounding the target pixel. In each box, enter the convolution multiplication value to use for that pixel. For example, to multiply the height value by 3 of the pixel immediately to the left of the target pixel, place the number 3 in the box immediately to the left of the center box.
4. Not all boxes need to be filled. Boxes left empty will cause the pixels they represent not to be used for the convolution calculation.
5. Enter the value for scale to be used. Convolution works by multiplying the height values of pixels used by the number in their box and then adding up the result. This result is divided by the scale number. Larger scale numbers result in less overall height.
6. Enter the value of offset, if any, to be used. The offset number is added to the height values computed as a result of the convolution divided by the scale. Positive numbers for offset increase overall height. Negative numbers decrease overall height.

See Also

Convolution filters for surfaces are very similar to convolution filters for images. See the Image - Filter topic for examples and tips for continuing education.
**Surface - Noise**

Add random height values. A rarely used command mainly of interest for simulations and creation of sample surfaces for testing surface processing scripts.

<table>
<thead>
<tr>
<th>Amount</th>
<th>Amount of random heights to add as a percentage of the total. The example above uses 6 percent.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preview</td>
<td>Check to see the modification in action.</td>
</tr>
</tbody>
</table>
**Surface - Tile**

Divides surface into square tiles of given extent and then averages pixel values within the tiles. This is used for aggregating and interpolating surfaces into larger tiles and for creating "pixelated" artistic effects. The tile transform operator is frequently used for this function. The tile command in the Surface menu exposes separate controls for X and Y tile dimension.

![Tile Surface Example](image)

- **X Tile**  
  Width (East - West extent) of tiles in pixels.

- **Y Tile**  
  Height (North - South extent) of tiles in pixels.

- **Use same X and Y**  
  Changes in either X Tile or Y Tile will update the other parameter. The illustration above uses 15 for both X Tile and Y Tile.

- **Preserve Values**  
  Use the closest existing value to the average within a tile.

- **Preview**  
  Check to see the modification in action.

**See Also**

See the Transform - Surface - Tile Operators topic for some nice terrain screenshots of tiled surfaces.
Surface - Transform

The Surface - Transform menu selection is enabled when the optional Surface Tools extension has been installed and the focus is on a surface window or on a surface layer in a map window. If you have not activated the Surface Tools extension with a valid Surface Tools serial number you will not be able to use the Surface - Transform command.

Launching the Transform command in the Surface menu launches the Transform dialog.

- **Scope**: The pixels in the surface that will be affected by the formula. Allows choice of [All Pixels], [Invisible Pixels] or any selection or saved selection made in the surface.
- **Surfaces**: A list of all surfaces in the project. Double-click on a surface name to add it to the Formula pane at the current cursor position.
- **Formula**: Allows entry of any formula that will transform the surface as desired. For details on formulas, operators and examples see the Transforming Surfaces topic.
- **Save result as new component**: If not checked (the default) the formula operates on the surface that had the focus when the dialog was launched, if the surface is not read-only. If checked, the system will first make a copy of the context surface and will then execute the formula in the context of that copy.
- **OK**: Execute the formula and make any changes directed to the surface.
- **Cancel**: Exit the dialog without making any changes.

The Scope specifies the pixels that will be affected by the transformation, and can be [All Pixels], [Invisible Pixels] or any selection or saved selection made in the surface. Formulas can contain numbers, names of surfaces (names of surfaces must be enclosed in square brackets []) if the name contains a space character, arithmetic operations and functions. Formulas are case insensitive. When we press the OK button, each pixel in the surface in the scope will be replaced with the value of the formula for that pixel.

If we would like to preserve the original surface unmodified we can check the Save result as new component box and the system will create a new surface (a copy of the surface which had the focus when the dialog was launched) and put the results of the formula there. The Save result as new component also allows us to...
surface transformations on a surface that is a read-only component, such as perhaps a surface that has been linked into a project from an Enterprise storage using the Enterprise Edition of Manifold System.

See Also

Surface Tools
Transforming Surfaces
Transform Dialog Functions and Operators
**Surface - Quantize**

Convert surface to the specified number of height values using the most frequently occurring heights.

<table>
<thead>
<tr>
<th>Values</th>
<th>Number of height values to be selected into which the total height range of the surface should be divided.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preview</td>
<td>Check to see the modification in action.</td>
</tr>
</tbody>
</table>

For a surface ranging from 0 to 100 meters choosing a **Level** of 8 will resample the surface to find eight height values that most closely approximate the range of height values in the original surface.

The screenshot above was created with 8 as the **Values** number. Because a relatively small number of pixels used the very highest heights the region of greater heights is approximately with a single height shown in green color. The great majority of the surface (not seen in the above cropped views) had height values in the ranges approximated by the magenta and blue shades so these were used for the other seven values.
Surface - Resize

Change the number of pixels in the surface using desired interpolation method.

<table>
<thead>
<tr>
<th>Method</th>
<th>Choose the interpolation method to be used to re-compute new pixel values as pixels are added or removed to achieve the new size.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Width</td>
<td>Width of the resized surface in pixels.</td>
</tr>
<tr>
<td>Height</td>
<td>Height of the resized surface in pixels.</td>
</tr>
<tr>
<td>Constrain Proportions</td>
<td>Checking this box will cause any change in Width or Height to automatically update the other parameter so that the width to height proportion is kept the same.</td>
</tr>
</tbody>
</table>

The size readout at the bottom of the dialog gives the size of the surface in kilobytes at the specified width and height.

The **Method** box chooses the mathematical technique used to interpolate the existing pixels into a greater or lesser number of pixels as the surface changes size. Different methods will at times work better or worse depending on the surface content. Choose from **Bicubic**, **Bilinear** and **Nearest Neighbor**. The default **Bicubic** method works best with most surfaces.

The screenshot above was created by deleting the least significant digit in the **Width** and **Height** values of the original surface, thus reducing the size of the surface by a factor of ten. Note that "size" in this case means the number of pixels used to represent the surface. A georegistered surface will continue to cover the same size on Earth after "resizing". If the surface is made one-tenth the height and width of the original, each individual pixel will simply cover the ground area that one hundred pixels formerly covered.
**Surface - Convert To**

Convert the numeric types used for height information in the surface.

<table>
<thead>
<tr>
<th>Type</th>
<th>Choose the numeric type to be used.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Autoscale Values</strong></td>
<td>If checked, will recompute the heights to fit in a proportional dispersion through the range of the selected type. For example, if converting from floating point to an integer type allowing -128 to +127 the height values will be rescaled to fit proportionally through this range.</td>
</tr>
<tr>
<td><strong>Embed missing values</strong></td>
<td>Enabled when Autoscale Values is checked. Reserve one value at the uppermost end of the range to be used to code invisible pixels that represent missing (invisible) values.</td>
</tr>
<tr>
<td><strong>Save result as new component</strong></td>
<td>Convert data and save as a new surface component in the project pane. When this option is checked, read-only surfaces (such as those linked in from an Enterprise server if Enterprise Edition is being used) may be converted.</td>
</tr>
</tbody>
</table>

The size readout at the bottom of the dialog gives the size of the surface in kilobytes using the selected numeric type.
Surface - Contours

The Contours dialog works with surfaces to create a drawing that includes areas representing contours from the surface.

To create contours:

1. Open a surface in a surface window, or click on a surface layer in a map.
2. Choose Surface - Contours from the main menu.
3. Supply a Name for the drawing to be created and (optionally) a description.
4. Choose the type of contours to be created in the Create box.
5. Specify the contour values to be used in the Height pane.
6. Press OK.

To specify values to be used in the Height pane:

1. Click the Add button.
2. Double click into the resulting new row and change the Height value to the value desired.
3. Organize the heights by using the Move Up / Down / to Top / to Bottom buttons.

Note that the Contour dialog refers to "heights" because most often in a GIS context it is used with surfaces that represent terrain elevation data. It can also be used to create contours with abstract data where the value in the surface represents some other value such as temperature or population.

<table>
<thead>
<tr>
<th>Name</th>
<th>Name for the drawing to be created that will contain the contours.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Optional description for the drawing.</td>
</tr>
<tr>
<td>Create</td>
<td>How contours will be created:</td>
</tr>
<tr>
<td></td>
<td><strong>Areas above heights</strong> - Create contour areas that show all locations higher than a particular contour height. Contour areas for higher heights will overlap contour areas for lower heights. This is the “wedding cake” effect.</td>
</tr>
<tr>
<td></td>
<td><strong>Areas below heights</strong> - Create contour areas that show all locations lower than a particular contour height. Contour areas for lower heights will overlap contour areas for higher heights. This is the “inverse wedding cake” effect.</td>
</tr>
<tr>
<td></td>
<td><strong>Areas between heights</strong> - Create contour areas that show all locations with height between the height intervals given. No contour areas will overlap. The most frequently used option when creating contour areas.</td>
</tr>
<tr>
<td></td>
<td><strong>Lines</strong> - Create lines at a given contour level. The classic</td>
</tr>
</tbody>
</table>

3417
contour line effect use within printed topographic maps.

**Heights Pane**
A list of heights to be used for contours. Can also be used to specify abstract values when a surface represents data other than terrain elevation.

- **Move to Top** - Move the highlighted height range to the top of the heights pane.
- **Move Up** - Move the highlighted height range up one position in the heights pane.
- **Move Down** - Move the highlighted height range down one position in the heights pane.
- **Move to Bottom** - Move the highlighted height range to the bottom of the heights pane.
- **Delete** - Delete highlighted height range from the contours to be created.
- **Add** - Add a new height range from the contours to be created.

**Example**

Visiting the USGS web site we’ve downloaded the **Montara Mountain** 1:24K-scale SDTS file that provides a DEM module for terrain elevation. We’ve imported it as a surface. Open the surface in a window.

Choose **Surface - Contours** to launch the **Contours** dialog.
In the **Contours** dialog click **Add** to add a new row in the **Heights** pane.

<table>
<thead>
<tr>
<th>Height</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000.00</td>
</tr>
<tr>
<td>3</td>
</tr>
</tbody>
</table>

Double click into the new row that's created and add a **Height** of 3.

<table>
<thead>
<tr>
<th>Height</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000.00</td>
</tr>
<tr>
<td>100</td>
</tr>
<tr>
<td>3</td>
</tr>
</tbody>
</table>

Click **Add** again and add a **Height** of 100.

<table>
<thead>
<tr>
<th>Height</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000.00</td>
</tr>
<tr>
<td>500</td>
</tr>
<tr>
<td>250</td>
</tr>
<tr>
<td>100</td>
</tr>
<tr>
<td>3</td>
</tr>
</tbody>
</table>

Click **Add** and add **Heights** at levels of 250 and 500 as well. Press **OK**.

The result is a new drawing in the project. If we open the drawing we see it consists of non-overlapping areas, where each area shows the contours between a given height range. The drawing's table has one field, the **Height** field, which gives the value of Height used to create each area.
We can thematically color the new drawing using thematic formatting controlled by the **Height** field, or simply by using the Color dialog to quickly color the areas with different colors.

For a nice effect, we can create a map using the original surface and then drag and drop the thematically colored contour drawing into the map in a layer above the surface. We then change the opacity of the drawing’s layer to 50%. This creates a visual mixture of the thematically formatted contours and the surface shading.

Another classic map can be created by choosing the **Lines** option in the **Create** box in the **Contours** dialog and to then layer the resultant contour lines in a map above the surface, or to use contour lines drawing in a map above a different drawing.

When the **Lines** option is chosen for the **Create** parameter, Manifold will add a **Height** numeric column to the table associated with the contour lines drawing with the height of each contour line in that column. This makes it easy to label contour lines with their height.
Terrain Menu

Terrain - Surface

The **Surface** dialog for terrains specifies how the data surface that defines the terrain should be visualized. Includes options such as the texture used for the surface, the color determined by the elevation, the level of details computed and the size of the data set in view.

### Surface Dialog Controls

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Palette</strong></td>
<td>Choose from preset color gradients that will be used to color the surface based on the elevation value.</td>
</tr>
<tr>
<td><strong>Apply</strong></td>
<td>Apply the chosen palette to terrain.</td>
</tr>
<tr>
<td><strong>Reverse</strong></td>
<td>Reverse the colors used in the values box from high to low.</td>
</tr>
<tr>
<td><strong>Interpolate</strong></td>
<td>Change the colors used in the values boxes by interpolating between the top and the bottom boxes. A quick way of creating smooth gradients of colors or sizes.</td>
</tr>
<tr>
<td><strong>Lighten</strong></td>
<td>Lighten all colors in the values boxes. Each click of the Lighten button lightens the colors a bit more.</td>
</tr>
<tr>
<td><strong>Darken</strong></td>
<td>Darken all colors in the values boxes. Each click of the Darken button darkens the colors a bit more.</td>
</tr>
<tr>
<td><strong>Grayscale</strong></td>
<td>Convert all colors in the values boxes to grayscale.</td>
</tr>
<tr>
<td><strong>Load from File</strong></td>
<td>Load a previously saved palette from an XML file.</td>
</tr>
<tr>
<td><strong>Save to File</strong></td>
<td>Save this palette to an XML file.</td>
</tr>
<tr>
<td><strong>Texture</strong></td>
<td>The texture applied to the surface, such as a variety of stone or other textures.</td>
</tr>
<tr>
<td><strong>Details</strong></td>
<td>Choose a level of detail to be computed, from <strong>Full</strong> to <strong>Lowest</strong>. Higher levels of details will require more computation and will result in slower display within the terrain window.</td>
</tr>
<tr>
<td><strong>Display</strong></td>
<td>The number of tiles to show in the terrain window. Terrains are subdivided into a smaller number of square tile regions, with each tile being 32 x 32 pixels. Operating with a larger number of tiles will increase the apparent extent of the surface in view in the terrain window. Increasing the number of tiles will slow down operation.</td>
</tr>
<tr>
<td><strong>Z scale</strong></td>
<td>Factor to apply in rendering height. A value of 0.50 will render heights half as high as their elevation values would suggest. A value of 2.00 renders heights twice as high.</td>
</tr>
<tr>
<td><strong>Walls</strong></td>
<td>Show the terrain with vertical &quot;walls&quot; at the edges of the terrain.</td>
</tr>
<tr>
<td><strong>Wireframe</strong></td>
<td>Show the terrain as a wireframe mesh instead of as a continuous surface.</td>
</tr>
<tr>
<td><strong>Preview</strong></td>
<td>Check this box to show the effect of options in the terrain window as they are altered.</td>
</tr>
</tbody>
</table>
Choose a collection of options, using **Preview** if desired to see the effects and then press **OK** to apply them to the terrain window.

> When choosing a new palette, don’t forget to press the **Apply** button to load the values from that palette.

**Keyboard Shortcuts for Z Scale in Terrain Windows**

With the focus on the terrain window we can use the following keyboard shortcuts to modify **Z scale**.

- **Page Up**  Increase Z scale by .01
- **SHIFT-Page Up**  Increase Z scale by .10
- **Home**  Set Z scale to 1.0
- **Page Down**  Decrease Z scale by .01
- **SHIFT-Page Down**  Decrease Z scale by .10

**Examples**

![Texture Example](image1)

![Texture Example](image2)

**Textures**: The upper image has no texture applied. The bottom image has a rough surface texture applied. Textures can provide a greater sense of reality.
Display: The three images above show an increasing number of tiles. The first image has the least number of tiles and shows terrain only in the foreground. The middle image adds more tiles, so the displayed terrain extends further towards the horizon. The last image has the greatest number of tiles used and so extends even further to the horizon. Tiles are square pieces of terrain that are 32 x 32 pixels in extent.

Details: This setting specifies what percentage of the data set is used to render the 3D terrain. The upper image uses Full details while the lower image uses Lower details. Lower Details settings will display faster but will simplify the terrain by using fewer data points from the terrain data set. Note how the lower image is an interpolated approximation to the upper image.

Palettes are a sequence of colors that are applied to the 3D surface depending on the elevation of the surface. Manifold provides numerous preset palettes.

<table>
<thead>
<tr>
<th>Elevation</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>540.0000</td>
<td><img src="image1.png" alt="Color" /></td>
</tr>
<tr>
<td>1006.0000</td>
<td><img src="image2.png" alt="Color" /></td>
</tr>
<tr>
<td>1436.0000</td>
<td><img src="image3.png" alt="Color" /></td>
</tr>
<tr>
<td>1922.0000</td>
<td><img src="image4.png" alt="Color" /></td>
</tr>
<tr>
<td>2380.0000</td>
<td><img src="image5.png" alt="Color" /></td>
</tr>
<tr>
<td>2338.0000</td>
<td><img src="image6.png" alt="Color" /></td>
</tr>
</tbody>
</table>

The palette used in the illustrations above will color the surface in a series of green colors from a lighter green to a green brown in higher elevations and then finally to a brighter green at the highest elevations.
Most of the terrain visible in these illustrations, like the screen shot above, is at elevations lower than the dark brown band and is therefore colored in the lower two bands.

Palette colors can be changed at will. For example, we can double click into the green color well just before the brown well and change it to bright yellow.

The result in the terrain window will be that terrain elevations from 1464 to 1922 will be colored bright yellow.

Different palettes may have greater or fewer divisions and color wells. Although many palettes are available to color terrains in a realistic way, some palettes are designed for use with purely abstract data.

These may be applied to terrains for otherworldly effects, or simply to color-code data shown in the terrain window.

 Wireframe
The **Wireframe** option shows the terrain as a wireframe mesh instead of as a solid surface. The colors of the wires will be taken from the palette color and texture for that part of the surface.

**Walls**

The **Walls** option turns on vertical panels that "close in" the area underneath the terrain and provide greater orientation when the terrain is seen from far away.
Terrain - Overlay

The Terrain - Overlay dialog allows us to overlay the terrain surface with images, objects from drawings, labels or other surfaces. Multiple images, drawings, labels and surfaces can be overlaid at the same time.

Images will be re-sampled and "spray painted" or "draped" onto the terrain surface. Objects from drawings will be converted into equivalent 3D objects and placed within the 3D terrain landscape. Colors used for drawing objects overlaid onto the terrain will be taken from their formatting in their home drawing.

To Overlay an image or other component onto a Terrain

1. Verify that the terrain and all components to be overlaid are correctly georegistered by viewing the terrain's surface and the components together in a map.
2. Open the terrain in a terrain window.
3. Choose Terrain - Overlay.
4. In the Components pane, check the boxes of the components to be overlaid. Use the move buttons in the dialog's toolbar to move components up and down to the order in which they are to be overlaid.
5. Choose options as desired for areas, labels, lines and points, and press OK.

The choices in the Terrain - Overlay dialog allow control over how drawing objects are represented by 3D objects in the terrain view.

- **Show All** - Show all available components overlaid on the terrain surface. Check all the boxes.
- **Show None** - Do not overlay any components on the terrain surface. Uncheck all boxes.
- **Show Inverse** - Invert checkboxes. Uncheck those previously checked, and check those that were not checked.
- **Move to Top** - Display the highlighted component uppermost when overlaid on the terrain.
- **Move Up** - Move the highlighted component up one position in the overlay stack.
- **Move Down** - Move the highlighted component down one position in the overlay stack.
- **Move to Bottom** - Display the highlighted component at the bottom of the overlay stack.

**Areas**

- **Embed into texture** - Show areas by embedding them into the textured surface of the terrain.
- **Paint over terrain** - Show areas by creating 3D area objects just above the terrain surface. Area objects will take their colors from formatting used in the drawing.
- **Opacity** - Opacity to use for areas drawn in the terrain. Enabled only when Paint over terrain option is checked.
- **Walls** - Check to display vertical "walls" which are the extruded vertical sides of areas shown in the terrain. Walls are drawn using the color specified for the area's border and using the Opacity specified for the area in this dialog. Enabled only when Paint over terrain option is checked.
- **Vary height by column** - Check to vary the altitude over the surface at which the area is drawn using the contents of a data field (column) called Height in the drawing's table. Enabled only when Paint over terrain option is checked. The height is reckoned over the highest...
location in the surface under the area. If the value of height is less than zero it is set to zero. The values in the Height column are assumed to use the same units as the surface.

Labels  Display options for labels:

Embed into texture - Show labels by embedding them into the textured surface of the terrain.

Paint over terrain - Show labels by creating 3D label objects just above the terrain surface. Label objects will take their style from the label style and colors used in the labels component. 3D labels will always face the viewer in the terrain window. To create 3D labels consisting of "text only" use a transparent background color for the labels in the labels component.

Size  - Relative size of labels.

Height  - Height in pixels above the surface to position labels

Spacing  - Interval in pixels between labels attached to lines.

Piers  - Check to display rod-like vertical supports under labels.

Lines  Display options for lines:

Embed into texture - Show lines by embedding them into the textured surface of the terrain.

Paint over terrain - Show lines by creating 3D line objects just above the terrain surface.

Size  - Relative thickness of lines.

Height  - Height in pixels above the surface to position lines

Smooth line body - Apply antialiasing to provide a smoother appearance for lines.

Piers  - Check to display rod-like vertical supports under lines at each coordinate defining the line.

Points  Display options for points:

Embed into texture - Show points by embedding them into the textured surface of the terrain.

Paint over terrain - Show points by creating 2D or 3D point objects just above the terrain surface.

Size  - Relative size of point symbol used to display points.

Height  - Height in pixels above the surface to position point symbol.

Style  - 2D icon or 3D shape to be used as a symbol for points.

Frame 2D Symbols - Draw a wireframe rectangle about 2D point symbols using the foreground color used for points in the component.

Texture Opacity  Opacity to use for the generated texture when embedding labels, areas, lines or points into the terrain. 60% by default. The partial transparency used by the default setting makes the embedded texture seem a part of the terrain. The same opacity value will be applied to all overlaid components.
**Preview**  Check to see how options will appear in the terrain window as they are chosen.

Check one of the display options (Embed into texture or Paint over terrain) to show areas, labels, lines or points in the overlay. If neither Embed into texture nor Paint over terrain is checked, then objects of that type will not appear in the overlay.

Overlays can be very slow to initially compute with graphics cards that do not support textures in hardware. Even in fast cards, the initial computation of an overlay can take a few minutes. Once the terrain redispays with the overlay, viewing will continue at the normal speed.

**Example**

We've created a map that has a labels layer over a surface layer created from the Montara Mountain sample surface from the Manifold CD.

![Map Image](image)

To add labels to a terrain we begin by opening the terrain.

![Terrain Image](image)

In the terrain we have navigated using keyboard navigation commands to a view over Pilarcitos Lake. The terrain has been colored using the Altitude palette. The view is toward the Southeast.
We open the Terrain - Overlay dialog and in the components pane we check the Labels component to be overlaid.

In the Labels portion of the dialog, we check the Paint over terrain box and uncheck the Embed into texture box. Press OK.

The result is that a label appears over Pilarcitos Lake just as it does in the map. The small vertical line under the label is the pier.
Note that when changes are made in the labels component those changes will be updated in the terrain window as well. For example, if we click on the labels layer in the map and change the formatting of the label so it uses blue background color and a label style that surrounds the letters in background color this change will also appear in the terrain window as well.

A good way to adjust the appearance of labels in a terrain window is to open the labels component in a second, small window and to make changes in label formatting in the labels window until the desired appearance is achieved in the terrain window. As the labels are changed in the labels window the terrain window will also be updated with labels in the same formatting.

Here is an example of labels embedded into the texture and shown with partial opacity.

Example
The choice of how lines are overlaid will greatly affect the appearance of lines from drawings that are shown in terrain windows. In this example, we have overlaid the Montara Mountain sample surface with roads taken from a 1:24K-scale USGS SDTS drawing.

Using **Embed into texture** and unchecking **Paint over terrain** the line appears as part of the surface texture of the terrain.

If we also check **Paint over terrain** we can simultaneously paint the line floating slightly over the terrain. We have lowered the **Texture opacity** to 20% in order to achieve a pseudo-shadow effect for the line embedded in the surface.

Note that painting lines over the terrain will result in sharper line rendering since this method is not constrained by the resolution of the terrain texture. Rendering overlaid lines by embedding them into the surface texture will often lower the resolution of lines because the texture cannot be rendered within any greater pixel resolution than that computed for the surface seen in the 3D terrain view.

Here we have unchecked the **Embed into texture** choice and have checked the **Piers** option. This option draws vertical supporting "piers" at each coordinate point that defines the line.
Finally, we can uncheck the Piers option to show just the line itself floating slightly above the surface. It is usually important to paint lines with some height above the surface so that the line does not intersect any interpolated bumps in the surface that would hide the line. Therefore, the default Height parameter is set to 50%.

**Pay Attention to Geographic Location**

Note that if the terrain is located in a different place than the images or drawings that are overlaid upon it, there will be no visual effect seen in the terrain window. This may seem to be an obvious comment; however, it is easy to forget that some components have been georegistered and others have not yet been georegistered. Check georegistration by showing the terrain together in a map with any other components that are to be overlaid upon the terrain.

Terrains can be georegistered by georegistering their associated images. Any image that is to be overlaid onto a terrain should also be georegistered. Although drawings are frequently imported from formats that automatically result in a georegistered component within Manifold, it is possible that drawings imported from CAD formats or other non-geographic formats are not yet georegistered. These should be georegistered before being overlaid onto a terrain that has already been georegistered.

**Notes on Resolution**

A common task is to overlay an image onto a terrain to give the terrain a more photorealistic look. Since it is usually the case that images used to overlay a terrain are not exactly the same resolution as the surface used to create the terrain, Manifold will automatically interpolate colors from the image's pixels to compute the required color for each individual surface pixel used to create the terrain.

Therefore, the maximum display resolution for an image overlaid upon a terrain will be the resolution of the surface used to create that terrain. If the surface is 1000 pixels by 1000 pixels and it is overlaid by an image that covers the same region in 10000 pixels by 10000 pixels, then each surface pixel will be colored by an interpolation of the 100 image pixels that fall within it. The resulting terrain will not appear to be "painted" with an image as detailed as the original image used for the overlay.

To allow use of greater detail from images that are higher resolution than the surfaces in use, use the Surface - Resize command to increase the number of pixels used by the surface. Alternatively, use a higher resolution surface covering the same region.

**See Also**

Overlays - For example of overlays of drawings and surfaces.
Area Overlays - Areas overlaid on terrains with 3D walls.
Terrain - Clouds

The terrain window can display synthetic "clouds" in the sky provide a greater sense of reality. The Terrain - Clouds dialog provides options that control the appearance of the clouds.

- **Show clouds**: Check this box to display clouds in the terrain view.
- **Quantity**: Choose a value for the relative number of clouds that appear in the sky.
- **Size**: Specify the relative size of clouds.
- **Opacity**: Specify how opaque clouds will be. Low opacity results in fainter, hazy clouds.
- **Preview**: Check to see how options will appear in the terrain window as they are chosen.
**Terrain - Fog**

Fog is an effect created in a terrain window by applying a color to an increasing amount to points located at an increasing distance from the viewer. The visual result is very similar to that of real haze or fog, which has a greater effect on distant points because they are seen through more haze. The **Terrain - Fog** dialog provides options to control the appearance of fog in the terrain window. Applying fog to a terrain enhances the sense of 3D realism.

- **Show fog**: Check this box to display fog in the terrain view.
- **Render fog from**: Show fog from the nearer percentage to the greater percentage of the visible range.
- **Color**: Color of fog to use. This is normally the same as the color of sky, a very light blue.
- **Preview**: Check to see how options will appear in the terrain window as they are chosen.

**Examples**

![Default fog](image1)

Default fog. A slight amount of fog applied to a terrain enhances reality by providing haze in the distance.

![Increased fog](image2)

We can increase the fog by increasing the nearer number in the **Render fog from** parameter. This provides a hazier landscape.

![Color change](image3)

Changing the color of fog changes the color of the sky as well, since sky is just fog color piled up to an infinite distance. Color shown is a dark gray color. Using black and a slight amount of fog creates night effects.
Terrain - Lighting
Terrain windows can provide relief effects as though the 3D terrain surface is illuminated from the side, causing highlights and darker regions in the surface. Lighting effects are usually required to provide a credible 3D simulation. Lighting options are set in the Terrain - Lighting dialog.

- **Show lighting effects**
  Check this box to show lighting effects in the terrain view.

- **Amount**
  Increasing the amount will increase the darkness of lowlights and the brightness of highlights.

- **Angle**
  Specify the direction from which the light originates. Angles are given in geographic compass directions.

- **Preview**
  Check to see how options will appear in the terrain window as they are chosen.

Examples

Without lighting effects the landscape lacks visual relief. The darker shades are caused by the palette, which in this example is set to color the surface with darker colors at higher elevations. Wise use of palettes can provide 3D visual cues and a more depth-like appearance, but not as much as lighting effects.

Applying lighting from the default South-East angle immediately provides depth and relief to the landscape. The highlights and darker tones appear as if a light source was located to the Southeast of our location. In this example, we are facing North in the terrain view.

Changing the Angle of illumination will at times result in dramatically different appearance. In this example we have changed the angle to North-East so that the low cliffs facing us down by the water become shadowed.
Increasing the Amount lighting provides a starker effect, with brighter highlights and darker shadows.

**Notes**

Lighting effects in the terrain window are a simulation that approximates the true physics of lighting. Darker regions are not true, cast shadows. Rather, they are regions of the surface that have been computed to face away from the light source and so are rendered darker. The approximation is nonetheless very credible and useful as a means of showing the 3D structure of the surface.

The Angle setting allows us specify lighting angles that are not possible locations for sun angles. For example, the terrain used in the example is a section of the Grand Canyon, which is located in the Northern Hemisphere in the Southern part of the United States. Sun angles at the Grand Canyon are usually more Southerly and are never North-East as shown. Using angles other than those possible as sun angles can be a good way of bringing out 3D nuances in the terrain surface that would otherwise be hidden.
Terrain - Water

Terrain windows can display a horizontal "waterline" plane cutting through the surface that is used to simulate the appearance of water in the landscape. The Terrain - Water dialog options specify the appearance of the waterline plane.

- **Show water**: Check this box to display water in the terrain view. Checked on by default.
- **Level**: Specify the water level from lowest to highest elevation value in the terrain.
- **Density**: Controls the "density" of the water. As water becomes denser it is more difficult to see through, especially from a viewpoint within the water.
- **Opacity**: Specify how opaque water will be. Low opacity results in transparent water that allows the submerged terrain surface to show through. Slight opacity provides a "depth" effect.
- **Color**: Click into the color well to change the color used for the water.
- **Preview**: Check to see how options will appear in the terrain window as they are chosen.

**Examples**

The default waterline plane appears as a river within the terrain.

Increasing the **Level** has the effect of moving the water level upwards and flooding the terrain. This is a topologically accurate effect so that waterline analysis may be done to determine visually which parts of the terrain are at the same level.
Decreasing **Opacity** causes the water to become more transparent and thus allow some of the terrain surface to show through. If we carefully compare this illustration and the previous one we can see that the water in the previous illustration was not fully opaque. The slight transparency of the water allowed part of the dark shadow of the headland to show through and thus give the water a sense of "depth."

We can change the color if a different water color is desired. In the example above we’ve changed color and transparency to achieve a muddy water effect, a more credible color in the case of flooded terrain.
Terrain - Snap to Surface

The **Snap to Surface** command toggles automatic snapping of the terrain camera to terrain-following mode, so that in response to any keyboard navigation commands the camera moves just slightly above the terrain elevation. The effect is somewhat like driving a car over the surface.

The view we see in a terrain window may be thought of the viewpoint of a camera located at a particular position over the terrain and aimed in a particular direction. Using the keyboard navigation keys specified in the Terrains topic, we can move the camera and point it in whatever direction we like.

Sometimes we would like to move the camera about the terrain while staying just above the terrain's surface, almost as if we were walking up and down the terrain with our eye-level just above the terrain surface. This can be very tedious to do using keyboard commands, so Manifold provides a terrain-following "automatic pilot" of sorts in the form of the **Snap to Surface** command.

When **Snap to Surface** is turned off, the camera flys about following whatever keyboard navigation commands we issue. For example, if the camera is pointed forward with no upward or downward tilt and we issue a **W** command to move forward the camera will move forward at the same altitude. If we tilt the camera down with the down arrow on the keyboard and then issue a **W** command the camera will move forward and downwards at the tilt angle used. With **Snap to Surface** turned off, we can freely "fly" the camera like an airplane.

When **Snap to Surface** is turned on the camera descends vertically down from whatever current location it is at and then responds to all keyboard navigation commands by moving just slightly above the terrain surface. Keyboard navigation commands to increase the elevation of the camera, such as the **Q** command to move vertically upward, are ignored. With **Snap to Surface** turned on, we can freely "drive" the camera over the surface like a car.

See Also

Terrains

Window Menu

The **Window** menu allows us to arrange windows within the main Manifold window.

- **New Window** - Enabled when a window is activated. Open a new window into that component.
- **Close** - Close the active window.
- **Close All** - Close all windows.
- **Next** - Activate the next window in the stack.
- **Previous** - Activate the previous window in the stack.
- **Cascade** - Arrange windows neatly in front of each other and offset slightly.
- **Tile Horizontally** - Arrange windows next to each other left to right and resize so they fill the main window.
- **Tile Vertically** - Arrange windows next to each other top to bottom and resize so they fill the main window.
- **Windows...** - Open a dialog that lists all windows.

See the Windows Dialog topic for additional info.
Window - Windows Dialog

The Windows dialog within the main Window menu shows a list of all windows open within the main Manifold window. We can use this dialog to activate (that is bring to the top) any window or to close several windows at once.

- **Activate** - Move the highlighted window to the top of the windows stack and make it the active window.
- **Close** - Close the highlighted windows.
- **Cascade** - Arrange windows neatly in front of each other and offset slightly.
- **Tile Horizontally** - Arrange windows next to each other left to right and resize so they fill the main window.
- **Tile Vertically** - Arrange windows next to each other top to bottom and resize so they fill the main window.

**Activate** Move the highlighted window to the top of the windows stack and make it the active window. Same as the Activate toolbar button and added to provide a similar look to the equivalent dialogs in Microsoft products.

**Close** Close the Windows dialog.

Example

Suppose we have four windows open showing various components in our project as seen above.
The **Windows** dialog would appear as above. Note that the **roads data** window is at the top of the list. The list shows the windows in their order in the windows stack. Each window listed will be shown with a small icon for the type of component it is.

We can click on the **roads data** window entry to highlight it and then **CTRL click** on the **SanFran** entry to highlight it as well.

If we now click the **Close** toolbar button both the **roads data** and the **SanFran** windows will be closed.

The **Windows** dialog will be updated with a new list of windows:
Tech Tip

Queries can be opened in two forms, either as a query window allowing the query text to be edited or in the form of a table window that reports the results of a query being run. As a convenience for users, if a query is open at the same time as an editable query window and also as a table window reporting the results of the query being run, the Windows dialog will append the postfix (Table) to the latter. For example, if we have a query called MyQuery and it is open both as an editable query window and also as a table window reporting the results of running the query, the Windows dialog will show two windows open, one called MyQuery and the other called MyQuery (Table).

Help Menu

Help - Activate

Each serial number that authenticates a license for Manifold System product has a limited number of activations. Therefore, it is critically important for users to read and understand the Activation Keys and Serial Numbers topic before using the Help - Activate command.

When first launched after installation, Manifold System raises the Activation dialog to allow entry of a serial number for preliminary functioning or the entry of both a serial number and an Activation key for permanent installation. At any time thereafter, the Help - Activate command may be selected to re-activate the system. Before using this command, it is wise to check the Help - About dialog to see what Manifold System edition and extensions have already been activated on the subject system.

Typical uses of the Help - Activate command include:

- Changing the serial number of a system to a different serial number during the preliminary installation period when running on serial number alone.
- Changing the serial number of a system to a different serial number after permanent installation (requires use of an Activation key).
- Installing a different Manifold edition, such as installing Enterprise Edition on a machine on which Professional Edition has already been installed. In that case, we would raise the Help - Activate dialog to allow entry of the new Enterprise Edition serial number and Activation key.

When Help - Activate is launched, the system will close any open projects and raise the Activation dialog to allow entry of a serial number or a serial number and an Activation key. The system will then shut down Manifold so that Manifold may be re-launched with fresh activation information.

A related dialog is the Help - Activate Extension dialog, which is used to turn on an optional extension such as Business Tools, Geocoding Tools or Surface Tools.

Note: The Help - Activate command is not used when a Manifold installation is running with license authorization from a Manifold License Server. In that case, the Help - Activate command has no effect on the License Server authorization because a License Server authorization always takes precedence over any local activation status.

Controls

Serial number  A serial number authenticating a Manifold System license. Serial numbers must match the release level of the Manifold System product. For example, after
installing Manifold System Release 7.00, a 7.00 Manifold System serial number must be used. A 6.50 Manifold System serial number won’t work.

**System ID**
An identifier for this particular installation. The System ID is automatically generated by Manifold and cannot be edited. It can be copied by highlighting the System ID and using `CTRL-C` or by right-clicking on the highlighted System ID and choosing **Copy** in the context menu.

**Activation key**
An Activation Key fetched using the serial number and System ID. Activation keys may be fetched automatically if the computer is connected to Internet, or they may be fetched manually. After a period of 30 days from the date the serial number was issued, an Activation key is required along with the serial number to activate Manifold.

**Accept**
Accept the serial number and (if entered) the Activation key.

**Cancel**
Abandon the activation process.

**Get Activation Key via the Web**
After entering a valid serial number, press this button to fetch an Activation key automatically if the computer is connected to the Internet. When this button is pressed, Manifold will take the **Serial number** provided together with the **System ID** and will connect to the Manifold key server via the web and request an Activation key for that **Serial number** and **System ID**.

**Caution:** getting an Activation key via the Web will use up one of your five activations. Do not fetch an Activation Key unless you are ready to immediately activate Manifold for permanent use on this computer.

If all five activations already have been used up, or if the serial number is not valid, getting an Activation key via the web will fail.

**Example**

John installed and activated a Manifold System Professional Edition license on a machine in his department several months ago. Since then, his company has acquired a few dozen Enterprise Edition licenses for installations and now installs Enterprise Edition on all machines. John would like to upgrade the one remaining Professional Edition machine to Enterprise Edition.

John acquires an Upgrade to Enterprise Edition product on the Manifold Online Store, trading in the Professional Edition license by providing the serial number for that Professional Edition license during the checkout process for his upgrade purchase. He gets an email with the new Enterprise Edition serial number for the upgrade.

John launches the Help - Activate dialog and enters the new Enterprise Edition serial number provided to him for his upgrade. He does not enter an Activation key. When he presses **Accept** Manifold shuts down. The next time John launches Manifold, the Activation dialog pops open to let him know how many days he has left in the preliminary installation period of his Enterprise Edition serial number. John clicks **Accept** to continue though the dialog to launch Manifold in the usual way. He sees that Enterprise Edition features have been enabled.

After a few days, John realizes there is no point in procrastinating and he may as well activate the Enterprise license on this machine, so the next time he launches Manifold and the Activation dialog pops open he presses the Get Activation Key via the Web button to automatically fetch an Activation key. He presses **Accept** and now Manifold is activated on that machine as Enterprise Edition.

**Note**

If you have several Manifold System licenses you can determine which serial number was used on a particular system as well as which extensions, if any, have been activated by launching the Help - About dialog.

**See Also**
Activation Keys and Serial Numbers
Installing and Activating a Manifold Extension
Help - Activate Extension

Each serial number that authenticates a license for a Manifold System extension product has a limited number of activations. Therefore, it is critically important for users to read and understand both the Activation Keys and Serial Numbers topic as well as the Installing and Activating a Manifold Extension topic before using the Help - Activate Extension command.

The Help - Activate Extension command is used to turn on the Business Tools, Geocoding Tools or Surface Tools optional extensions to Manifold System. When Help - Activate Extension is launched, the system will close any open projects and raise the Extension Activation dialog to allow entry of a serial number or a serial number and an Activation key. The system will then shut down Manifold so that Manifold may be re-launched with fresh activation information. When Manifold is re-launched it will automatically be reconfigured so that the activated extension will be fully enabled. Before using this command, it is wise to check the Help - About dialog to see what Manifold System edition and extensions have already been activated on the subject system.

There is no need to specify which extension is being turned on, since each serial number contains coded information specifying the extension that it activates. For example, if you license the Surface Tools package from manifold.net you will receive a Surface Tools serial number. Entering that serial number into the Extension Activation dialog will automatically turn on Surface Tools.

Just like the serial number used to enable Manifold, serial numbers for extensions typically allow a temporary period of usage, normally 30 days, without requiring an Activation key. This preliminary operation period allows time for users in remote areas to acquire an Activation key while still being able to use the extension. While an extension is operating on serial number alone the Help - About dialog will report how many days are left on that serial number before an Activation key is required.

Controls

<table>
<thead>
<tr>
<th>Control</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Serial number</td>
<td>A serial number authenticating a license for the desired extension. Serial numbers must match the release level of the host Manifold System installation. For example, when installing Surface Tools in a Manifold System 7.00 Professional Edition installation, a Manifold 7.00 Surface Tools serial number must be used. A 6.50 Surface Tools serial number won’t work.</td>
</tr>
<tr>
<td>System ID</td>
<td>A “fingerprint” for this particular machine’s hardware and software configuration that uniquely identifies this computer system. The System ID is automatically generated by Manifold and cannot be edited. It can be copied by highlighting the System ID and using CTRL-C or by right-clicking on the highlighted System ID and choosing Copy in the context menu.</td>
</tr>
<tr>
<td>Activation key</td>
<td>An Activation Key fetched using the serial number and System ID. Activation keys may be fetched automatically if the computer is connected to Internet, or they may be fetched manually. After a period of 30 days from the date the serial number was issued, an Activation key is required along with the serial number to activate an extension.</td>
</tr>
<tr>
<td>Accept</td>
<td>Accept the serial number and (if entered) the Activation key.</td>
</tr>
<tr>
<td>Cancel</td>
<td>Abandon the activation process.</td>
</tr>
<tr>
<td>Get Activation Key via the Web</td>
<td>After entering a valid serial number, press this button to fetch an Activation key automatically if the computer is connected to the Internet. When this button is pressed, Manifold will take the Serial number provided together with the System ID and will connect to the Manifold key server via the web and request an Activation key for that Serial number and System ID. Caution: getting an Activation key via the Web will use up one of your five activations. Do not fetch an Activation Key unless you are ready to immediately activate the extension for permanent use on this computer. If all five activations already have been used up, or if the...</td>
</tr>
</tbody>
</table>
serial number is not valid, getting an Activation key via the web will fail.

**Note**

If you have several Manifold System licenses you can determine which serial number was used on a particular system as well as which extensions, if any, have been activated by launching the Help - About dialog.

**See Also**

 Activation Keys and Serial Numbers
 Installing and Activating a Manifold Extension
Help - About
In addition to copyright and trademark information the Help - About dialog lists the following information about the your Manifold System installation:

- **Edition** - The edition of Manifold System installed, such as Professional Edition or Enterprise Edition.
- **Build / Version Number** - Provide this number to technical support if you ever contact technical support in connection with a support incident.
- **Installed extensions** - Provides a list of optional extensions, such as Business Tools, Geocoding Tools or Surface Tools that have been enabled via the Activate Extension dialog. Extensions activated with a serial number alone will report how many days are left on the serial number. Extensions activated both with a serial number and a permanent Activation key will not show a time to expiration.
- **Serial Number** - The serial number used to install this license. The last few digits are concealed with "X" characters to keep your full serial number secret. You can use the leading digits to determine which serial number was used to install this license if you have several serial numbers. The "X" masked version of the serial number cannot be used to fetch activation keys - this is to protect your serial number from unauthorized use.
- **CPU** - The central processing unit reported by Windows.
- **GPU** - Whether the graphics processing unit is CUDA capable or not.
- **RAM** - The amount of dynamic RAM memory reported by Windows as installed in the computer. Note that 32-bit versions of Windows may report less than the full amount of physical RAM actually installed.
- **Operating system** - The Windows version and build reported by Windows.
- **32-bit or 64-bit mode notice** - This status line indicates whether Manifold is running in 32-bit mode or 64-bit mode.

Manifold Build / Version Numbers

Manifold build numbers (sometimes referred to as version numbers) have four parts separated by dots, for example, 7.1.0.67. We may read this example as follows:

- The first two parts identify the **product version** so that 7.1 indicates release 7.1, the immediate successor to release 7.00. For marketing purposes, 7.1 series product releases are collectively known as the 7x generation.
- The third part identifies the **patch number**. Patch numbers are incremented to indicate any updates issued after initial release. In the example, a 0 indicates this is the original release of this product iteration and is not a subsequent patch.
- The fourth part is a revision number used mainly within manifold.net to identify various internal candidate builds created on the way to publication of new product version and patch for public use. In the example this is revision 67.

Note that Manifold build / version numbers have changed with 7x and subsequent product releases. The old system used a single number, such as "Build 429" which increased linearly. The new system provides a four-part nomenclature to help keep track of more frequent builds and more frequent updates.

See Also

Help - Activate
Activate Extension
Activation Keys and Serial Numbers
Installing and Activating a Manifold Extension
NVIDIA CUDA
Help - Check for Updates

The **Automatic Updates** page in the Tools - Options dialog sets up Manifold to automatically check via Internet for any updates that may have been issued to the currently running release.

The **Check for Updates** command provides a manual way to check for updates. Launching the command tells Manifold to connect to the manifold.net update server, checks to see if there have been any updates issued for the currently running version of Manifold and, if so, will pop open a message inviting the operator to visit the update site.

No information is ever transmitted to the update server and no data except information on published updates is ever downloaded from the update server.

This command requires Internet access for Manifold using ordinary HTTP protocol, the same as used by web browsers like Internet Explorer. If the computer is not connected to Internet or if HTTP is blocked by a firewall or other security mechanism, this command cannot work. In general, if you can launch a web browser and visit web sites from your computer this command will work.

**Installing Updates**

Many customers do not like complex software to alter their systems, so Manifold updates must be consciously downloaded and installed. In addition, some updates such as major new releases may not be free but may require payment of a license fee.

The automatic update notification system will notify you when a new update is available and will offer to launch the web page for that update. You will only be able to view that web page if your computer is connected to Internet and you have an Internet browser, such as Internet Explorer or other browser, installed on your computer.

Read the updates web page carefully for instructions on downloading the update, uninstalling your previous Manifold installation and then installing the update.

**See Also**

**Automatic Updates** in Tools - Options
Help - Contents

Manifold Help uses the standard Microsoft Help system or a web-based browser version if a local Help file has not been installed. Selecting the Help - Contents command launches Help set to the Contents tab, which displays the top level of the Help "outline" structure of topics, showing the main list of Help "books," or chapters.

The Contents command is the first choice for either reading Help in sequential fashion or for drilling down through the chapters to find a topic that is (presumably) arranged in logical order within the hierarchy. We can then explore Help by following hyperlinks (shown in boldfaced underlined green text like those in the See Also section below) to read related topics, discover definitions of Manifold terms of art and so on.

Help is a very large document, with thousands of pages of information. There's no hope of learning Manifold efficiently if we do not learn to make savvy use of Help facilities such as Contents, Index and Search.

Installing Help

Manifold displays the web-based version of Help by default. To install a local version that may be used if your computer is not connected to Internet, please see the How to Install this Documentation topic.

See Also

Help - Index
Help - Search
Help - Index

Manifold Help uses the standard Microsoft Help system. Selecting the Help - Index command launches Help set to the Index tab, which displays a list of all text words and phrases that have been indexed by the authors of Help.

The Index command is a good way to find top-level, reasonably major words and phrases of interest. Entering the first letters of a word or phrase will scroll the list of entries to those that lexicographically match those letters. Clicking an entry and pressing the Display button will launch the Topics Found dialog that lists all topics in which that indexed entry is found.

Because it has been compiled by actual humans, Index tends to have fewer "hits" than the Help - Search function. The downside is that it is not as relentlessly comprehensive as the machine-compiled Search list. The usual order is to check the Index first and then second to use the Search function via the Find tab.

Help is a very large document, with thousands of pages of information. There's no hope of learning Manifold efficiently if we do not learn to make savvy use of Help facilities such as Contents, Index and Search.

See Also

Help - Contents
Help - Search
Help - Search

Manifold Help uses the standard Microsoft Help system. Selecting the Help - Search command launches Help set to the Find tab, which launches access to a machine-created comprehensive database of words and, optionally, phrases.

Using Search is very comprehensive, but can result in a very long list of topics for common GIS words compared to the Index command. When looking for a particular term, the usual order is to check the Index first and then second to use the Search function via the Find tab.

Help is a very large document, with thousands of pages of information. There's no hope of learning Manifold efficiently if we do not learn to make savvy use of Help facilities such as Contents, Index and Search.

See Also

Help - Contents
Help - Index
Help - Manifold on the Web

The Help - Manifold on the Web menu launches an Internet browser session to connect to a variety of web sites of interest to Manifold users. You must have a functioning Internet connection and an installed Internet browser to use this capability.

Community The Manifold online community, the Georeference worldwide forum of Manifold users voluntarily participating in an Internet forum for discussion and mutual assistance.

Free Stuff The Free Stuff page on the manifold.net web site. A collection of free scripts and programming examples, including IMS example web sites.

News The News page on the manifold.net web site.

Online Support The Support page on the manifold.net web site, a useful first stop for general support information, service packs and other information of interest to technical support.

Manifold Home The manifold.net home page.

Note that Internet pages for the internal links in the Manifold on the Web menu are kept current for the duration of the current Manifold release. If the Manifold release covered by this documentation has been replaced by a newer release, the Internet pages may change.

The Manifold Online Community

The Community choice connects to the online Georeference forum at http://www.georeference.org run by Manifold users. Note that this is not a "factory" web site managed by manifold.net - it is simply an online meeting place where Manifold users from all over the world can meet to discuss Manifold System.

The Community forum can be a great place to discuss Manifold with your friends online, to give and to get technical assistance and to talk about new ideas for Manifold and for using Manifold. Experienced members of the Manifold online community offer the following advice when visiting the forum:

- People who read and make posts in the forum are ordinary users just like you. They all have their "day jobs" and are visiting the forum on their own time.
- There is a zero tolerance policy on insults, ad hominem attacks, foul language and other incivilities. Please keep it polite, respectful of other participants and professional. Be patient with beginners.
- If you need help using Manifold, don't be impatient to get answers to your postings. Remember, other people reading the forum are ordinary users just like you. If they have time and take an interest in helping you, they may post a reply. But any participation by anyone is completely voluntary. If you have a technical question and need a response for sure, see the Technical Support topic.
- People in the online Manifold community tend to be very generous with their time unless they feel they are being taken advantage of. Postings of the form "I don't have any time to learn about this... someone please do it for me" won't usually get many responses. In contrast, a posting that shows you have studied the documentation and have tried to work out a problem before raising it in the forum will usually get many responses from all over the world.
- When engaging someone in a discussion thread, do not email them directly unless invited to do so by that person. Any responses to postings in the forum should be made on the forum where everyone can read them and where people have a choice as to whether they want to respond or not.
- As in any community, your reputation in the community depends upon you. If you are courteous and knowledgeable and helpful you will find many supportive users who will reciprocate. Keep it clean and technical. Do not use profanity and do not engage in ad hominem attacks on others. If you find yourself commenting on someone else instead of on product technology, you are crossing the line.
- Keep in mind that users from all over the world participate in the online Manifold community and that English is not a first language for many participants. Sometimes it is helpful to use simple English and to avoid slang if a language gap arises.
- manifold.net does not moderate the forum nor monitor it for suggestions. If you have a suggestion to improve the product, read the Contacting manifold.net topic and then send your suggestions directly to manifold.net. Use the forum as a good place to discuss your ideas with other Manifold users so that when you send in your suggestion it is as strong as possible.
• Like any unmoderated, online community that anybody can join, although the forum has many well-informed users with expert skills it also has its share of self-appointed experts who have poor skills and a poor grasp of technology. The real experts will usually call out errors, but they cannot catch every error. Don't believe everything you read on this or any other forum. If you need authoritative information, hire an expert or utilize a manifold.net tech support service product.

• There may be people who are employed by manifold.net and who are active in the forum. If so, they are there on their own time as ordinary hobbyists with a personal interest in using Manifold. They are not paid to participate in the forum, so don't think they have any special obligations to answer questions or to provide a special communications channel to manifold.net. If anything, other members of the Manifold online community will be especially annoyed at newbies who pester any manifold.net employees on the forum because they know that is the surest way to drive such people off the forum. Postings from any manifold.net employees on the forum do not represent the views of manifold.net - only direct communications from manifold.net on the website should be taken as official information.

• The Manifold online community is for discussing Manifold products. It is not for gossip about Hollywood, current events or personalities, or for religious, political, financial, legal, medical or other off-topic posts. Do not post advertising or spam in the forum. Do not seek or give advice such as medical advice, legal advice or other regulated professional opinions or advice. For example, no essays about how to install water heaters in a way that hopefully will not electrocute anybody or gas the occupants of a dwelling. Please stick to technical discussions of Manifold products and reasonably closely associated other software technologies. Although the forum is not moderated, a blacklist mechanism provides a way of eliminating spammers, habitual rule violators, sociopaths or other folks who just can't stick to a productive, professional, courteous technical discussion of Manifold products.

• See the forum for updated guidelines and additional information.

See Also

Contacting manifold.net
Technical Support

Toolbars

Toolbars

Manifold duplicates many menu commands in the form of toolbar buttons. Buttons all have Tool Tips, which are quick labels that pop open if the mouse cursor lingers on the button. Toolbars may be turned ON and OFF under the Tools - Customize menu. A right mouse click on the main menu bar calls up a shortcut to the Tools - Customize menu.

Individual toolbars may be dragged to different locations on the display. They may be "docked" on any of the screen edges or left "floating" in the middle of the display. Move a docked toolbar by dragging the 3D ridge at the left or top of the toolbar. Move a floating toolbar as you would any window, by dragging its title bar. Use a CTRL-drag to drag a toolbar past the normally docked position. On exit, Manifold remembers how you left your toolbars positioned.

Manifold toolbar buttons are context sensitive and are not enabled unless they can be applied. For example, unless we are working on an image the Select Pixels selection mode button will not be enabled in the Selection toolbar.

Toolbars know what type of component is the active window and can be customized to provide a different set of buttons for different components. For example, the Shapes toolbar provides a different set of commands for drawings than it does when an image is the active window.

The toolbars are shown below in their undocked form. Toolbars may be undocked by dragging them (using the small 3D ridge at the left side or top of the toolbar) into the main Manifold window. In docked form, toolbars do not have a title bar.

To quickly see which toolbar is which, use Tools - Customize to turn it ON and OFF. As the toolbar turns ON and OFF it will be obvious which it is.
Alignment Toolbar - Aligns and sizes controls in forms and aligns objects in drawings.

Format toolbar - Controls the formatting in drawings, images and labels layers. The drawing version shown formats foreground/background color, pattern style and pattern size for areas, lines and points. The image and labels versions show foreground and background color, style and size.

Navigation toolbar - Zoom, center and pan functions.

Selection toolbar - Select objects and/or pixels. Shows selection mode, selection filter, mouse cursor method and choice of styles by which selection is shown.

Snap toolbar - Specifies snap modes for use in drawings. Snap to forces the motion of the mouse cursor to jump between allowed locations only.
Tools toolbar - Toolbox for creating objects in drawings, for painting into images or entry of text labels. The drawings and images forms of the toolbar are illustrated. This toolbar will change to suite whatever is the active type of component (that is, drawing or image or text label component).

Tools (Advanced) toolbar - Toolbars used with forms include the default Tools toolbar and the Tools (Advanced) toolbar. The Tools toolbar for frequently used controls is used with Forms to create Windows standard user interface controls. The toolbar is enabled when a Form is the active window. Forms are used to provide user interfaces to scripts, to view tables and for other functions within Manifold. Advanced controls are used with Forms to create Windows standard user interface controls. The toolbar is enabled when a Form is the active window. Forms are used to provide user interfaces to scripts, to view tables and for other functions within Manifold.

Tracing toolbar - Works within maps to create objects in drawings by automatically tracing over images. This function is called vectorizing in traditional GIS packages.

Transform toolbar - A toolbar loaded with operators that can accomplish "one step" editing and transformation of the contents of the active window. The transform toolbar's operators box will be loaded with the applicable operators for whatever type of window is active (drawing, image, etc.).

Tech Tip

The default installation of Manifold System contains the toolbars described in this documentation. If toolbars are missing or do not appear in the Tools - Customize dialog, check the Config folder within the Manifold System installation folder for an XML files that may have been added containing <addin> entries. An add-in may be used to add commands to Manifold and also to control the availability of toolbars.
Alignment Toolbar

The **Alignment toolbar** is most frequently used to neatly align controls such as text boxes and command buttons in forms. It can also neatly align objects in drawings. Finally, it is used to align elements in print layouts. When used with drawings, the alignment toolbar is most frequently used in CAD style drawings such as diagrams of office or factory layouts.

The alignment toolbar is turned off by default. To turn it on, open a component that uses the alignment toolbar (such as a layout), choose Tools - Customize and check the box for the alignment toolbar.

To use the alignment toolbar, click on the form or drawing that has objects to be aligned and then press the alignment control desired. If no selection is present, the alignment toolbar works on all objects. If a selection is present, the alignment command will apply only to the selected objects or controls.

Some alignment commands require a **primary** selected object that is taken as the reference point.

![Alignment Toolbar Image]

- **Align Left** - Move objects so their left edges are aligned to the left edge of the primary selected object.
- **Align Top** - Move objects so their top edges are aligned to the top edge of the primary selected object.
- **Align Right** - Move objects so their right edges are aligned to the right edge of the primary selected object.
- **Align Bottom** - Move objects so their bottom edges are aligned to the bottom edge of the primary selected object.
- **Center Horizontally** - Move objects so that their centroid is centered horizontally in the form. In a drawing, this centers the centroid within the bounding box of all objects in the drawing.
- **Center Vertically** - Move objects so that their centroid is centered vertically in the form. In a drawing, this centers the centroid within the bounding box of all objects in the drawing.
- **Space Across** - Space objects evenly across the form in a horizontal direction. In a drawing, this spaces the centroid of each object evenly across the bounding box of all objects in the drawing.
- **Space Down** - Space objects evenly up and down the form in a vertical direction. In a drawing, this spaces the centroid of each object evenly up and down the bounding box of all objects in the drawing.
- **Same Width** - Resize objects so that the width of all is the same as the width of the primary selected object.
- **Same Width (Max)** - Resize objects so that the width of all is the same as the width of the widest object.
- **Same Width (Min)** - Resize objects so that the width of all is the same as the width of the narrowest object.
- **Same Height** - Resize objects so that the height of all is the same as the height of the primary selected object.
- **Same Height (Max)** - Resize objects so that the height of all is the same as the height of the widest object.
all is the same as the height of the tallest object.

**Same Height (Min)** - Resize objects so that the height of all is the same as the height of the shortest object.

**Full Width** - In layouts, increase width of this element to the full page width, within borders. In drawings, increase width of objects to the full width of the bounding box of all objects.

**Full Height** - In layouts, increase height of this element to the full page height, within borders. In drawings, increase height of objects to the full height of the bounding box of all objects.

**Full Size** - In layouts, increase width and height of this element to the full page size, within borders. In drawings, use the size of the bounding box of all objects.

When using **Align Top** and **Align Bottom** it's easy to command the system to stack up all the objects on top of each other. Simply click and drag them to different positions.

**See Also**

See the Align Items in Layouts topic for an example of aligning text labels in a print layout.
Edit Toolbar

The Edit Toolbar provides commands used to edit objects in drawings. If not already displayed, the edit toolbar will appear when an object is selected for editing with a 
CTRL-ALT clicking the object.

The most frequently used edit toolbar commands have keyboard shortcuts that allow them to be toggled on, switched and toggled off with a single key when an object is selected for editing. Using the keyboard shortcuts will speed up work flow in many situations because the mouse cursor can be kept near the work while the keyboard is used to rapidly choose the command desired. For example, with an object selected pressing the A key (either uppercase or lowercase) chooses the Add Coordinate command. Pressing the D key will switch from Add Coordinate to the Delete Coordinate tool. Pressing the D key again will toggle the Delete Coordinate command off. Note that pressing the next command’s key will switch to that command without any need to first toggle off the previous command.

Edit Toolbar Commands Illustrated
The following sequence of illustrations show edit toolbar commands in action. Each is preceded by the toolbar icon and brief description. Illustrations show the action of the command in a simple situation so that the action is clear. Following is an enlarged view of the command’s toolbar icon with comments intended to help remember and recall the mnemonic nature of the icon. Illustrations use line objects. The same commands work areas as well.

**Add Coordinate (A)**

Add Coordinate (A) - Add a new coordinate wherever the tool is clicked, inserting the new coordinate into the segment nearest the clicked location.

Choosing this tool and then clicking near a segment of a selected object will add a new coordinate and edit handle to the object at the location of the click. Note that the coordinate is added not necessarily exactly on the segment, but at the location of the click. The click must be made close to the segment so that it is unambiguous to which segment the coordinate is to be added. If the coordinate is to be moved farther away from the previous segment, create it with a click, toggle the Add Coordinate command off and then drag the coordinate to the location desired.

The icon for this command shows a new coordinate (the black square) added and so the old line segment (the red segment) moves into the new black line shape.

**Add Coordinate on Segment**

3460
**Add Coordinate on Segment** - Add a new coordinate exactly on the segment nearest the clicked location.

This command is similar to the one above except that when clicking near a segment it creates the new coordinate exactly on the old segment at the location nearest the click. It is used in a variety of ways, including to provide specific targets for *Snap* modes or using *Snap* modes to add coordinates at intersections between objects.

The icon shows a new coordinate (the black square) added with no change in the line shape since the new coordinate is added exactly on an existing segment.

**Delete Coordinate (D)**

**Delete Coordinate (D)** - Delete the next clicked coordinate.
Clicking on a coordinate / edit handle with this tool deletes it, collapsing the two segments it separated into a single segment between adjacent surviving coordinates.

The icon shows the two line segments deleted (the red apex lines) when a coordinate at the apex is deleted. The result is the black line seen.

**Delete Coordinate / Split Branch**

Delete the next clicked coordinate together with both adjacent segments, treating the remnants to either side of the deleted segments as two branches. Enabled when there are sufficient coordinates so that deleting a coordinate can create two branches.

This tool is similar to the above, except when the clicked coordinate is deleted both the coordinate as well as the two adjacent segments are deleted, leaving a gap between what have become separate branches.
The icon shows the two line segments deleted (the red apex lines) when a coordinate at the apex is deleted, with no segment replacing them. The two black lines shown are the result.

**Split Branch**

Split Branch - Split the object into two branches at the next clicked coordinate.

Clicking this tool on a coordinate will split the object into two branches at that location. Note that there is no immediate difference in appearance, since the two objects are exactly coincident at their mutual ends at the clicked position, with no gap between them.

We can see this by **CTRL-SHIFT** clicking and dragging one of the branches to move it in its entirety to a slightly different position.
This shows that there really were two branches formed. In reality, we will often split an object into two branches not to simply move one of the branches but rather to delete one of the branches. This is faster than deleting many coordinates one by one to delete the undesired portion of the object.

The icon shows a gap where the branch is split at the apex to indicate it is split into two branches. Note that in reality no gap appears when one branch is split into two branches since the ends of both branches will be exactly coincident at the location of the split.

**Delete Branch**

- **Delete Branch** - Delete the clicked branch. Enabled when there is more than one branch.

Available only for objects consisting of more than one branch, click this tool on any coordinate of a branch to delete that branch.

The icon shows a red figure where the entire branch is deleted. Note that in reality if a line consists of a single branch this last branch cannot be deleted.

**Extend (E)**
Extend (E) - Extend the edited line to another line or area object, if by extending the last segment of the edited line it is possible to intersect the target object. Choose the tool, click the end of the line being edited and then click the object to which the line is to be extended.

We begin with a line selected that is to be extended so that it precisely meets the line to the upper right. We select the tool and click once on the line segment or end coordinate / edit handle of the line to be extended...

...and then we click on the line to which it is to be extended.

The last segment is extended in its current direction so that the last coordinate is precisely on the target line.
Note that this command does not "split" the target line. If we select the target line for editing, we can see that it consists of two coordinates. There was no new coordinate inserted at the intersection with the extended line.

The icon shows one line selected (with a black square) being extended to a more solid line that is also selected.

**Note:** Manifold provides a CTRL command to extend to the next intersection if the line already intersects the object. When using the **Extend** command, CTRL-clicking the object to which the line should be extended will extend the edited line to the next intersection with the object if the line already intersects the object.

**Leave (L)**

**Leave (L)** - Cut an edited object with another object, leaving the designated part. Choose the tool, click a location in the part of the edited object that is to be retained and then click the object to be used as a cutter.

Given a line selected for editing that is to be cut by a second line, we choose the tool and then we first click on the portion of the edited line to leave...
...and then we click on the cutting line.

The result is the edited line is cut by the cutter and any portion other the portion clicked is discarded.

The icon shows a cutting line cutting the selected line, leaving the portion originally clicked (with the black square) and deleting the portion in red on the other side of the cutter.

**Trim (T)**

Trim (T) - Cut an edited object with another object, trimming off the designated part. Choose the tool, click a location in the part of the edited object that is to be removed and then click the object to be used as a cutter.

To use Trim we first click on the portion of the edited line to be discarded, or trimmed...
...and then we click on the line to use as a cutter.

The portion of the edited line first clicked is removed, that is, trimmed.

The icon shows a cutting line cutting the selected line, deleting the portion originally clicked (with the black square and red line) and saving the portion in black on the other side of the cutter.

**Split (S)**

.Split (S) - Split an edited object with another object. Choose the tool and then click the object to be used as a cutter.

Given a line selected for editing that is intersected by some other line we click on the cutting line.
The line selected for editing is then split into two branches.

We can see it has been split into two branches because there are now two coincident terminal coordinate points located exactly at the location where the cutting line intersected the edited line.

The icon shows a black line being cut into two by a cutting line. Note that in reality a line split by a cutter will not have a gap at the location of the cut, since the ends of the two split pieces will be exactly coincident.

**Scroll Wheel**

When using a mouse with a scroll wheel we can use the scroll wheel to zoom in and out. This works even in the middle of commands such as the edit toolbar editing commands. By default the zoom will be centered on the mouse position. Holding the CTRL key while operating the scroll wheel will force the zoom to be near the center of the opened window.
Format Toolbar

Used with drawings or drawing layers in maps, the Format toolbar controls the color, style and size of areas, lines and points. Used with text components the Format toolbar controls the color and appearance of text labels in that layer. Used with images, the Format toolbar controls the color, style and size of painting tools. Note that the contents of this toolbar depend on what component window or map layer is active.

Images

The Format toolbar appears with images as a shortcut to settings in the Tool Properties pane. It is used to control the action of painting tools.

- **Foreground color** - The main color applied by the painting tool.
- **Background color** - Used with image editing effects where background color replaces some or all pixels.
- **Style** - Brush pattern or other tool style.
- **Size** - Size in pixels of the applied brush or other painting tool.

Drawings

- **Areas Format** - Foreground / background colors, Style and size for all areas in this drawing.
- **Area Borders Format** - Foreground / background colors, Style and size for all area borders in this drawing. While the 1/20th of a point value for border size works well for screen display it is wise to increase the thickness of area borders for printed materials since 1/20th of a point results in lines that are too thin for good visibility with most printers.
- **Lines Format** - Foreground / background colors, Style and size for all lines in this drawing.
- **Points Format** - Foreground / background colors, Style and size for all points in this drawing.

See the Formatting and Formatting Drawings topics for more information and detailed examples of use with drawings.

Themes
The format toolbar for themes appears the same as for drawings, except that any formatting characteristics that are inherited from a parent drawing or theme are shown with corner brackets. The first format toolbar illustrated above shows formatting for a theme that has no inherited characteristics and the second format toolbar illustrated above shows a theme in which all characteristics are inherited.

The above format comes from a theme in which all formatting characteristics for lines are inherited from a parent, while area and point formatting characteristics have been specified within the theme.

**Labels**

Formatting applied to a labels component (either in a labels window or in a labels layer in a map) applies throughout the entire component or layer.

**Font** - Choose a font from those installed on this Windows system.

**Size** - Font size.

**Bold** - Boldface the font.

**Italic** - Switch the font to italic (slanted) style.

**Underline** - Underline the font.

**Align Text Left** - Align text with an even left margin.

**Center Text** - Center each text line in the label.

**Align Text Right** - Align text with an even right margin.

**Label Foreground** - The color used for the body of text letters.

**Label Border** - The color used for borders in label styles that use borders.

**Label Background** - Used as fill color in label boxes, shadows and other complementary color.

**Style** - Label style.

**Label Rotation Angle** - Sets rotation of labels, either in fixed degrees or under the control of a field value.

**Align Left** - Place labels to left of tie points.
Center Horizontally - Center labels horizontally on tie points.

Align Right - Place labels to right of tie points.

Align Top - Place labels above tie points.

Center Vertically - Center labels vertically on tie points.

Align Bottom - Place labels below tie points.

Alignment controls (Align Text Left, Center Text, Align Text Right and Justify) format multiple-line text to left margin, centered or right margin in the text object box.

The other Align controls adjust the position of the label object box relative to the tie point for the label.

The left-center-right and top-center-bottom buttons can be combined, for example, to create a right-bottom alignment.

See the Labels topic for more on text labels.

Print Layouts - Text Boxes

Formatting applied to text box objects in a print layout applies to that object only. The text box must be selected for editing with a CTRL-ALT click to enable the format toolbar for that box.

Font - Choose a font from those installed on this Windows system.

Size - Font size.
**Bold** - Boldface the font.

**Italic** - Switch the font to italic (slanted) style.

**Underline** - Underline the font.

**Foreground color** - Used for body of letters.

**Background color** - Used as fill color in text box. Invisible color is used by default.

**Align Text Left** - Align text with an even left margin.

**Center Text** - Center each text line in the text box.

**Align Text Right** - Align text with an even right margin.

**Justify** - Align text with even left and right margins.

**Align Top** - Align text to top of text box.

**Center Vertically** - Center text vertically in text box.

**Align Bottom** - Align text to bottom of text box.

The **Align** and **Center** controls adjust the position of the text vertically and horizontally within the label box:
The left-center-right-justify and top-center-bottom buttons can be combined, for example, to create a right-bottom alignment.

If you are interrupted or must otherwise put this book down, you should take steps to renumber your current position in the text: count the words already read and jot the number in the margin. You can...

When only a few words are available, the Justify control does not create an elegant effect. It is much better if it has more words to work with on each line as seen above.
Navigation Toolbar

Use the buttons in the Navigation toolbar to zoom and pan within the drawing, image or map.

- **Back** - Go back one view in the current component window.
- **Forward** - Go forward one view in the current component window.
- **Zoom scale combo** - Choose a zoom scale from a variety of even zoom scales.
- **Zoom To Fit** - Zoom so that the component fits within the current window size.
- **Zoom In** - Magnify the view as if seen from a closer distance.
- **Zoom Out** - Reduce the view as if seen from farther away.
- **Zoom Box** - Zoom to the size of the cursor box drawn with the mouse.
- **Center Point** - Pan the view so that the spot clicked is centered.
- **Grabber** - Interactive pan: click and drag with the grabber hand. The scene will be panned so that the initial point is moved to the spot where the drag is released.
- **Tracker** - Provides interactive distance and area measurements. See the Tracker topic.

When zooming with images, zoom in and zoom out will zoom to discrete zoom levels. To zoom to a particular level, use the **View - Zoom To** menu command.

The **Back** and **Forward** buttons maintain a separate queue of views for each component window that is open.

Additional Mouse Buttons

Manifold component windows (such as a drawing, image or map window) will automatically work with standard commands issued by additional mouse buttons on mice so equipped, such as the IntelliMouse Explorer. For example, the fourth and fifth mouse buttons typically issue **Back** and **Forward** commands for views.
Query Toolbar
The Query toolbar provides fast selection in tables, drawings and drawing layers in maps using "one line" simple queries using data fields associated with objects.

Field box  Operator box  Value box
The Query toolbar consists of three boxes, from left to right: a field box, the operator box, and a value box. The example show selects all records where Place_name contains the string "DURANGO".

Field box
Also known as the column box. The field that will be queried. The illustration above shows that the Place_name field will be queried. The field box will show all fields available in the table, including intrinsic fields and any fields brought in via relations.

Operator box
The query operator to use. Press in the not button to the left of the operator box to use the "not" version of the operator. For example, pressing in the not button in the example above will find all records Not Equal to the given value.

Value box
The value to be used with the operator. Not all operators require values. For example, Median does not take a value in the value box.

We use dark blue, black, and violet bold face fonts in this topic to distinguish the three boxes to make examples and explanation more clear. In "real life," Manifold uses the same black font color in all three Query toolbar boxes.

To find objects or records using the query toolbar:

1. Open the drawing, table or map. If using a map, click on the drawing layer desired.
2. Choose the field to query in the field box.
3. Choose an operator in the operator box. Push in the not button if the "not" form of an operator is desired.
4. Provide a value in the operator box if one is required for the operator used.
5. Press Select.

All objects in a drawing or records in a table selected by the query will appear in red selection color.

Query Toolbar Operators

Operators are available depending upon the type of table column selected. For example, operators using tokens work only with text columns. Comparison operators such as Greater than will work with text, binary, coordinate system and geometry columns as well as with numeric columns. When used with non-numeric columns a comparison operator will compare data going byte-by-byte stopping on the first different pair of bytes.

Binary columns, coordinate system columns and geometry columns support Duplicates, Duplicates Except First, Equal To, Greater Or Equal To, Greater Than, Less Or Equal To, Less Than, Not Equal To, and Uniques operators.

Areas
Select records with geometry for area objects. Available for geometry columns.

Bottom
Find the lowest N values of the given field. For example, Bottom 5 will find records with the lowest five values of
the given field.

**Containing** Select records in which the given field contains the given string.

**Containing Match** Select records in which the given field contains the given regular expression.

**Containing Token** Select records in which the given field contains the given token. See the Transform - Using Tokens and Text Strings topic for information on tokens.

**Duplicates** Select records that contain duplicate values in the given field.

**Duplicates Except First** Select records that contain duplicate values in the given field, excluding the first record of each set of duplicates.

**Empty** Select records with empty values. Available for binary, coordinate system, geometry, text and URL columns.

**Ending with** Select records in which the given field ends in the given string.

**Ending with Match** Select records in which the given field ends in the given regular expression.

**Ending with Token** Select records in which the given field ends in the given token.

**Equal to** Select records in which the given field is equal to the given value.

**Greater than** Select records in which the given field is greater than the given value.

**Greater or Equal to** Select records in which the given field is greater than or equal to the given value.

**Less than** Select records in which the given field is less than the given value.

**Less or Equal to** Select records in which the given field is less than or equal to the given value.

**Lines** Select records with geometry for line objects. Available for geometry columns.

**Matching** Select records in which the given field matches the given regular expression.

**Maximum** Select records in which the given field is the maximum value of all values found in the table.

**Median** Select records in which the given field is the median value of all values found in the table.

**Minimum** Select records in which the given field is the minimum value of all values found in the table.

**Not Equal to** Select records in which the given field is not equal to the given value. Equivalent to pressing the *not* button and using Equal to.

**Points** Select records with geometry for point objects. Available for geometry columns.

**Sounding Like** Find records with strings that have similar pronunciation in English using the Soundex algorithm. For example, in the *Mexico_eg.mif* sample drawing selecting **Place_name Sounding Like doorangoh** will find the province of Durango.
Starting with  Select records in which the given field starts with the given string.

Starting with Match  Select records in which the given field starts with the given regular expression.

Starting with Token  Select records in which the given field starts with the given token.

Top  Find the highest N values of the given field. For example, Top 5 will find records with the highest five values of the given field.

Typical  Find records in which the value of the given field is within the given number of records of the median.

Uniques  Select all records for which the value of the given field is not duplicated in other records. Equivalent to using not with the Duplicates operator.

Examples

The following examples are based on the mexico_eg.mif sample drawing.

<table>
<thead>
<tr>
<th>Field</th>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Place_name</td>
<td>Sounding Like</td>
<td>doorangoh will select the province of Durango.</td>
</tr>
<tr>
<td>Area(I)</td>
<td>Top 5</td>
<td>will select the 5 largest areas.</td>
</tr>
<tr>
<td>Place_name</td>
<td>Duplicates</td>
<td>selects all area objects that have duplicate names. For example, there are three objects with a Place_name of CAMPECHE and two that are named SONORA. This is a fast way of finding area objects, such as provinces, that may be drawing using more than one area. This is typically done to show islands or other non-contiguous parts of a province, country, state or county.</td>
</tr>
</tbody>
</table>

Regular Expression Examples

The Matching, Containing Match, Ending with Match and Starting with Match operators use Visual Basic style regular expressions. Examples using mexico_eg.mif and the Place_name field would be:

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Matching</td>
<td>DURANGO selects one record (&quot;DURANGO&quot;),</td>
</tr>
<tr>
<td>Matching</td>
<td>CHI.+ selects two records (&quot;CHIHUAHUA&quot; and &quot;CHIAPAS&quot;),</td>
</tr>
<tr>
<td>Matching</td>
<td>.+OR[AE].* selects three records (&quot;SONORA&quot;, &quot;SONORA&quot; and &quot;MORELOS&quot;),</td>
</tr>
<tr>
<td>Containing Match</td>
<td>C[PQR] selects one record (&quot;VERACRUZ&quot;),</td>
</tr>
<tr>
<td>Containing Match</td>
<td>[^ A-Z] selects one record (&quot;EDO. DE MEXICO&quot;),</td>
</tr>
<tr>
<td>Starting With Match</td>
<td>BA.+ selects four records (all starting with &quot;BAJA&quot;),</td>
</tr>
<tr>
<td>Starting With Match</td>
<td>BB.+ selects nothing,</td>
</tr>
<tr>
<td>Starting With Match</td>
<td>[F-H] selects three records (&quot;HIDALGO&quot;, &quot;GUANAJUATO&quot; and &quot;GUERRERO&quot;),</td>
</tr>
<tr>
<td>Ending With Match</td>
<td>&quot;...&quot; (a space followed by three periods without quotation marks) selects three records (&quot;BAJA CALIFORNIA SUR&quot;, &quot;QUINTANA ROO&quot; and another &quot;QUINTANA ROO&quot;).</td>
</tr>
</tbody>
</table>

Token Operator Examples
Operators such as **Containing**, **Ending with** or **Starting with Token** use "tokens" which are groups of characters set off by separator characters. **Tools - Options** allows specifying what characters should be considered to be separators.

Suppose we have a field in a table listing the following set of processors:

- AMD K6
- AMD Athlon
- Intel 486
- Intel 486SX
- Intel Pentium
- Intel PII
- Intel PIII

Examples of different operators and results would be:

- **Containing Token** PII selects "Intel PII".
- **Containing** PII selects "Intel PII" and "Intel PIII".
- **Containing Token** 486 selects "Intel 486".
- **Containing** 486 selects "Intel 486" and "Intel 486SX".

**Tech Tip**

Suppose we would like to select all records with a null value in a given field. That is, we want to find all records that don't have a value in that field. There are at least five methods of doing so using the query toolbar. In the examples that follow below, do not type the " quotation marks in the value box. The field **Name** will be searched and all records with non-empty values in **Name** will be selected.

Use a regular expression requiring non-empty text:

- **Name Matching** ".+" (period and plus sign in the value box)

Use a regular expression with the **NOT** button:

- **Name Not Matching** ".*" (period and asterisk in the value box)

A hack using the **Sounding Like** operator:

- **Name Sounding Like** " " (nothing in the value box)

A hack using the **Less Than** operator:

- **Name Sounding Like** " " (space character in the value box)

An elegant use of the **Minimum** operator:

- **Name Minimum**

**Themes and Toolbars**

**Themes** can be used in the query and Transform toolbars. Object sets defined on themes will always use the name of the theme's parent drawing. For example, if there are selected objects in a theme named T whose parent is a drawing named D the selection choice will appear in the query or transform toolbar boxes as **Selection in D**.
This is done so that it is immediately clear that modifying a theme will modify the parent drawing (and thus all other themes bound to that drawing).

See also

The transform toolbar is also used to make spatial selections using commands such as Select Touching or Select Contained within that work between sets of objects.
Selection Toolbar

The Selection toolbar provides controls for **Selection Modes**, **Selection Filters**, **Selection Methods** and **Selection Style**.

### Selection Modes

The selection mode buttons control how a new selection made with a selection tool is combined with any existing selection. One selection mode button is always pushed in.

- **Select Replace** - Replace the existing selection with whatever is now selected.
- **Select Add** - Add whatever is now selected to the existing selection.
- **Select Subtract** - Subtract whatever is now selected from the existing selection.
- **Select Invert** - Deselect whatever is now selected that was already in the existing selection and otherwise add what has just been selected.
- **Select Intersect** - Select only the region of overlap between what is specified by the tool and what is in the existing selection. If there is no overlap, nothing will be selected.

### Selection Filters

Choose the type of items that are to be selected by pushing the selection filter buttons for the desired types. The buttons above show selection enabled for points and disabled for areas, lines, pixels and text labels. If we drew a selection rectangle over a region in the map that contained all four types of items, only the points within the rectangle will be selected. Areas, lines and pixels within the rectangle will be ignored.

- **Select Areas** - Enable selection of areas by mouse selection tools.
- **Select Lines** - Enable selection of lines by mouse selection tools.
- **Select Points** - Enable selection of points by mouse selection tools.
- **Select Pixels** - Enable selection of pixels by mouse selection tools.
- **Select Text** - Enable selection of text labels by mouse selection tools

All four filter buttons may be used in maps. When working with drawings, the **Select Pixels** button will be disabled. When working with images, the **Select Areas**, **Select Lines** and **Select Points** buttons will be disabled.
disabled. When working with images, **Select Pixels** is always enabled and will be shown in a "forced" style that cannot be clicked out.

### Selection Methods

These selection tool buttons specify how the mouse is used to select items in view. Only those items enabled by the Selection Filter buttons will be selected. Some selection commands come in regular and on-center versions. See the Selection topic for illustrated examples of the difference between the on-center version and the regular version of the selection tools.

- **Select Touch** - Click to select pixels like the one clicked, give or take the tolerance that may be set in **Tools - Options**.
- **Select Shape** - Select objects within a region defined by a series of clicks followed by a right click.
- **Select Freeform** - Select objects within a region defined by clicking and dragging, followed by a right click.
- **Select Box** - Click and drag to select items within a rectangular selection box
- **Select Box on Center** - Click and drag open a selection box that’s always centered on the point of first click.
- **Select Circle** - Click and drag open a selection circle within the rectangle implied by the mouse motion.
- **Select Circle on Center** - Click and drag open a selection circle that’s always centered on the point of first click.
- **Select Ellipse** - Click and drag open a selection ellipse within the rectangle implied by the mouse motion.
- **Select Ellipse on Center** - Click and drag open a selection ellipse that’s always centered on the point of first click.

**SHIFT key**  Shifts selection to "open" mode. Selects all items any part of which is within the shape drawn with the mouse.

### Selection Style

- **Selection Filter** - Appears with tables. Show only selected records in the table.
- **Selection Style** - Change the selection style to a less intrusive pattern or color or other style. The pull-down arrow next to the **Use Selection Color** button opens a menu of possible selection styles.
- **Select None** - Deselect all items.

See the main Selection topic for more information and detailed examples.
Snap Toolbar

Checking a choice in the Edit - Snap To menu controls Snap modes, which work with mouse cursor motion. Snap modes force the mouse cursor to take effect only at those discrete, specified locations that are nearest to the mouse cursor's location. See the Edit - Snap To topic for details and examples.

The snap toolbar is turned off by default. To turn it on, open a component that uses the snap toolbar (such as a drawing), choose Tools - Customize and check the box for the snap toolbar.

Snap Modes in Drawings and Themes:

- **Graticule**
  - Snap to graticule intersections if View - Graticule is enabled.

- **Grid**
  - Snap to grid intersections, if View - Grid is enabled.

- **Areas**
  - Snap to the coordinates that define areas.

- **Lines**
  - Snap to the coordinates that define lines.

- **Points**
  - Snap to points.

- **Segments**
  - Snap to arbitrary locations on segments defining line or area objects.

- **Selection**
  - Snap only to those objects in the selection.
  - Requires snap to areas, lines or points to be enabled.

  **TAB**
  - Depressing the TAB key jumps the cursor immediately onto the snap reticule. This is used when we want the mouse cursor to jump immediately to the snap point without a mouse click.

  **Space Bar**
  - Depressing the space bar turns off snap temporarily until the space bar is pressed again.
  - This is very useful to temporarily turn snap off.

Snap Modes in Layouts:

- **Grid**
  - Snap to grid intersections, if View - Grid is enabled.

- **Elements**
  - Snap to the corners of an element.

  **TAB**
  - Depressing the TAB key jumps the cursor immediately onto the snap reticule. This is used when we want the mouse cursor to jump immediately to the snap point without a mouse click.

  **Space Bar**
  - Depressing the space bar turns off snap temporarily until the space bar is pressed again.
  - This is very useful to temporarily turn snap off.

Snap Modes in Images, Labels, Profiles and Surfaces:
Graticule  
Snap to graticule intersections if View - Graticule is enabled.

Grid  
Snap to grid intersections, if View - Grid is enabled.

TAB  
Depressing the TAB key jumps the cursor immediately onto the snap reticule. This is used when we want the mouse cursor to jump immediately to the snap point without a mouse click.

Space Bar  
Depressing the space bar turns off snap temporarily until the space bar is pressed again. This is very useful to temporarily turn snap off.

Changing Snap Action with TAB and Space Bar

TAB  
Depressing the TAB key jumps the cursor immediately onto the snap reticule. This is used when we want the mouse cursor to jump immediately to the snap point without a mouse click.

Space Bar  
Depressing the space bar turns off snap temporarily until the space bar is pressed again. This is very useful to temporarily turn snap off. When snap is turned off using the space bar the status bar will not show snap modes enabled.

Pressing the Tab key will cause the mouse cursor to jump immediately to the snap cursor.

Snap Tolerance

The Snap Tolerance parameter may be optionally specified in the Tool Properties pane to control how close the mouse must be to a snap item before a snap mode takes effect.

When a snap mode is on, the Snap Tolerance setting gives the distance in screen pixels or in physical units within which the cursor must be to a given snap item (such as a line when snapping to lines) before the cursor snaps to that item.

When the cursor is outside the snap tolerance range, snaps are not effective. Setting a small snap tolerance is therefore a good way of retaining the benefits of snap modes without having the snap cursor jump about in a disconcerting way if the mouse is nowhere near a snappable item.

Layer Snap Restrictions

Using Layer Restrictions a map layer can be made non-snappable. When a layer is so restricted, any snap modes will ignore items in that layer.
See Also

Edit - Snap to for examples.
Editing with Snap
Editing Forms
Autocomplete with ALT
Tool Properties
Tools Toolbar
The commands in the Tools toolbar are used to insert new points, lines and areas into a drawing or to paint regions of pixels into images. Objects may be created "free hand" or pre-built shapes may be used. Note that the contents of this toolbar depend on what component window or map layer is active.

Drawings

The Tools toolbar for drawings hosts drawing editing tools. The difference between on-center and regular versions of these tools is exactly analogous to the difference between on-center and regular version of these tools used for images. See the Image Editing Tools topic for examples of the difference between regular and on-center versions of painting tools.

<table>
<thead>
<tr>
<th>Tool</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Insert Area</td>
<td>Draw areas within the mouse cursor region defined by a series of clicks followed by a right click.</td>
</tr>
<tr>
<td>Insert Freeform Area</td>
<td>Draw areas within the mouse cursor region defined by clicking and dragging, followed by a right click.</td>
</tr>
<tr>
<td>Insert Freeform Line</td>
<td>Draw lines as defined by clicking and dragging, followed by a right click.</td>
</tr>
<tr>
<td>Insert Line</td>
<td>Draw lines as defined by a series of clicks followed by a right click.</td>
</tr>
<tr>
<td>Insert Line Sequence</td>
<td>Draw lines as a sequence of separate lines.</td>
</tr>
<tr>
<td>Insert Point</td>
<td>Draw points - click to create a point.</td>
</tr>
<tr>
<td>Insert Box</td>
<td>Draw a rectangular box shape with a click and a drag.</td>
</tr>
<tr>
<td>Insert Box on Center</td>
<td>Draw a rectangular box shape centered on initial mouse click.</td>
</tr>
<tr>
<td>Insert Circle</td>
<td>Draw a circle with a click and a drag..</td>
</tr>
<tr>
<td>Insert Circle on Center</td>
<td>Draw a circle centered on initial mouse click.</td>
</tr>
<tr>
<td>Insert Ellipse</td>
<td>Draw an ellipse with a click and a drag..</td>
</tr>
<tr>
<td>Insert Ellipse on Center</td>
<td>Draw an ellipse centered on initial mouse click.</td>
</tr>
<tr>
<td>Insert Geographic Circle</td>
<td>Draw a shape centered on the initial mouse click that forms a circle on the Earth. That is, each location on the shape will be the same distance from the center point. May appear to be a non-circular shape depending on the projection of the drawing.</td>
</tr>
<tr>
<td>Create Lines</td>
<td>Create lines when drawing shapes.</td>
</tr>
<tr>
<td>Create Points</td>
<td>Create points when drawing shapes.</td>
</tr>
</tbody>
</table>

See the Adding Shapes topic for additional details and examples of working with drawings.
If Snap modes are not on, the ALT key toggles between non-center and on-center version of whatever tool is selected.

### Images

The Tools toolbar for images hosts painting tools. See the Image Editing Tools topic for examples of the difference between regular and on-center versions of painting tools.

- **Paint Areas**: A mode button: the selected tool will create filled regions of pixels.
- **Paint Lines**: A mode button: the selected tool will create lines.
- **Paint Points**: A mode button: the selected tool will create points where the mouse cursor changes direction on clicks.
- **Paint Area**: Paint pixels within the region clicked. Creates lines or points if shape modes so specify.
- **Paint Freeform**: Paint pixels within region about which the mouse is clicked and dragged.
- **Paint Line**: Paint a line defined by straight-line segments between clicks.
- **Paint Box**: Paint a rectangular box using a mouse click and a drag.
- **Paint Box on Center**: Paint a rectangular box centered on initial mouse click.
- **Paint Circle**: Paint a circle using a mouse click and a drag.
- **Paint Circle on Center**: Paint a circle centered on initial mouse click.
- **Paint Ellipse**: Paint an ellipse using a mouse click and a drag.
- **Paint Ellipse on Center**: Paint an ellipse centered on initial mouse click.
- **Brush**: Paint a dot with one click or a continuous line with a click and a drag.
- **Paint Bucket**: Click to fill a contiguous or enclosed region with color. Fills to a given threshold of color.
  - SHIFT click to pour into the given color in all regions of the image, even if not contiguous. Use this as a "replace color" operation.
  - CTRL click to replace colors within the threshold while retaining the same intensity.
  - CTRL and SHIFT may be combined to replace non-contiguous colors while retaining intensity.
- **Gradient**: Click, drag and release to create a color gradient between click point and release point.
- **Pick Color**: Click to pick the color at that location as the foreground color.
  - Shift click to pick the color at that location as the
background color.

See the Image Editing Tools topic for additional information on painting into images.

If Snap modes are not on, the ALT key toggles between non-center and on-center version of whatever tool is selected.

**Layouts**

**Insert Horizontal Line**  Create a horizontal line in the layout. Default line size is 1/20th of a point.

**Insert Vertical Line**  Create a vertical line in the layout. Default line size is 1/20th of a point.

**Insert Box**  Create a rectangular box with a click and a drag. Boxes have foreground and background color and line thickness that can be set by selecting the box and using the format toolbar. Default line size is 1/20th of a point.

**Insert Box on Center**  Create a rectangular box centered on initial mouse click. Default line size is 1/20th of a point.

**Insert Text**  Add a text box, such as a caption, copyright notice or title, to the layout. Click on the text box to select it and then use the format toolbar to format the text. Double-click a text box in the layout to edit the text it contains.

**Insert Legend**  Add a legend to the layout, based upon a component used in the layout.

**Insert North Arrow**  Add a North arrow to the layout, based upon a component used in the layout.

**Insert Scale Bar**  Add a scale bar to the layout, based upon a component used in the layout.

See the Layouts topic for additional information on creating and using layouts.

**Scripts**

Scripts provide programming instructions used to customize Manifold or to create new capabilities. See Programming Manifold and Scripts for an introduction to scripts.
If the Manifold Debugger is installed, when a script has the focus the tools toolbar will have debugger commands enabled. The debugger helps us develop and debug scripts in the scripting language of our choice. Commands will be enabled in the debugger toolbar as make sense given the execution state of the script.

- **Run**: Run a script.
- **Run under Debugger**: Run a script under control of the debugger.
- **Pause**: Pause execution of a script.
- **Stop**: Stop execution of a script.
- **Step Over**: Step over a routine.
- **Step Into**: Step into a routine.
- **Step Out**: Step out of a routine.

### Labels

The tools toolbar has three commands available when a labels component has the focus. These commands are used to manually add labels to an unbound labels component. An unbound labels component is one in which label text is not automatically created from fields for all objects in a parent drawing but rather a labels component in which labels are added manually. Labels components are normally used within maps so that when adding labels by hand there are other components visible (such as a drawing) to provide correct positioning context.

- **Insert Label**: Add a text label to the labels component.
- **Insert Line Label**: Add a text label aligned to the guideline composed of straight line segments that is drawn with the insert tool.
- **Insert Freeform Line Label**: Add a text label aligned to the freeform, curved guideline that is drawn with the insert tool.

See the Adding Text Labels topic for examples of the above.

### Forms

The **Tools toolbar** is used with Forms to create frequently used Windows standard user interface controls. The toolbar is enabled when a **Form** is the active window. Forms are used to provide user interfaces to scripts, to view tables and for other functions within Manifold.

Click on a control in the Tools toolbar and then click and drag in the Form window to insert the control in the form in the position and size indicated by the mouse click and drag. Forms are then customized using the **properties** for each control as well as the **View - Info Window** for the **Form**.

See the individual controls topics for details on each control.
Insert Check Box - Inserts check box controls.

Insert Option Button - Inserts option button controls. Only one option button in a group may be selected.

Insert Command Button - Inserts command button controls.

Insert Frame - Draws frame with caption. Frames are used to set off groups of controls, often option buttons.

Insert Text Box - Inserts text box. Also called an edit field or edit control. This control can display text entered by the programmer at design time or entered by the user or assigned to the control by other code at run time.

Insert Static Text Box - Add a customized text box.

Insert List Box - Inserts list box controls.

Insert Combo Box - Inserts combo box controls.

Insert Horizontal Scroll Bar - Inserts horizontal scroll bar controls.

Insert Vertical Scroll Bar - Inserts vertical scroll bar controls.

See the Tools (Advanced) Toolbar topic for additional controls used with forms.

See Also

Debugger

Layouts
Tools (Advanced) Toolbar

The **Tools (Advanced) toolbar** is used with **Forms** to create frequently used Windows standard user interface controls that are technically more advanced or less frequently used than those in the **Tools toolbar**. The toolbar is enabled when a **Form** is the active window. Forms are used to provide user interfaces to scripts, to view tables and for other functions within Manifold.

Click on a control in the Tools (Advanced) toolbar and then click and drag in the Form window to insert the control in the form in the position and size indicated by the mouse click and drag. Forms are then customized using the **properties** for each control as well as the **View - Info Window** for the **Form**.

See the individual controls topics for details on each control.

---

**Insert Animation** - Plays silent .avi files to create dialog effects like the sheet of paper that flies between folders in the Windows copy progress dialog. Can also play silent .avi files dropped onto the control. Manifold provides 19 cool Windows dialog .avi files to get started.

**Insert Chart** - Inserts a chart control.

**Insert Image List** - Contains a collection of images that can be used by other Windows Common Controls such as ListView, TreeView, TabStrip and Toolbar controls, as well as with other controls with a Picture property. Saves development time by maintaining all images in a standard, consistent catalog of images.

**Insert Image Combo Box** - Similar to standard combo but with ability to include pictures with each item in the list portion of the combo, with special facilities for list management.

**Insert Date / Time Picker** - Displays date and/or time info and provides an interface for modifying date and time. Dropdown menu provides a MonthView calendar.

**Insert Month View** - An easy way to view and set date information using a monthly calendar. Can select one or multiple dates or show up to 12 months at a time.

**Insert List View** - Displays data as ListItem objects with an optional icon. Four different views provide data as icons, small icons, a list or a report. A sophisticated control.

**Insert Progress Bar** - Create a progress bar display that can be used to show the progress of a process.

**Insert Masked Text Box** - Used to prompt users for input using a fixed format specified by a mask pattern, for example, a telephone number in a specified format. If a mask pattern is not used, this control behaves about the same as a standard text box.

**Insert Rich Text Box** - Allows entry and editing of text using advanced formatting features, such as paragraph formatting with left and right indents, hanging indents, bold and italic font and so on. Also supports object embedding using the OLEObjects collection. Write your own word processor or create elaborate forms for users.
Insert Slider - Insert a slider bar control that can be moved left and right.

Insert Tree View - Insert a hierarchical tree display. Designed to display data such as organization trees, entries in an index, files and directories on disk, etc.

Insert UpDown - Insert an increment/decrement control, also called spin buttons.

Insert Tab Strip - Insert a tabbed control where clicking on each tab brings it to the fore.

Insert Picture Clip - Displays graphics from bitmap, icon, metafile, JPEG or GIF files.

Insert Multimedia Control - Manages Media Control Interface (MCI) devices like sound boards, MIDI sequencers, CD-ROM drives, audio players, videodisc players, etc. Add voice note recording to your dialog, or play spoken announcements.

Insert Status Bar - Insert a status bar like those at the bottom of many program windows. Complete with a nearly infinite set of properties for the intrepid programmer.

Insert Tool Bar - Insert a toolbar at the top of the form that hosts buttons or other controls. Yet another control with a vast array of properties.

Insert Cool Bar - Requires installation of Internet Explorer 3.0 or greater. Provides a modern, "railbar" look and the ability to create (get this!) user configurable toolbars like those in IE.

Insert Common Dialog Control - Provides a standard set of Windows dialog boxes for opening and saving files and selecting colors and fonts. Can also display Help.

Insert System Info Control - Detects system events such as desktop resizing, resolution changes, time changes. Also provides operating system platform and version information and changes in AC/battery power status and Plug/Play hardware configuration. Used when writing applications for portable devices (GIS in the field) and for developing info for your tech support team.

Insert ActiveX Control - Insert any ActiveX control available on this system.

See Also

Tools Toolbar
Tracing Toolbar

The Tracing toolbar works within maps to create objects in drawings by automatically tracing over images. This function is called vectorizing in traditional GIS packages.

Tracing toolbar buttons are enabled when a drawing layer in a map is clicked. Tracing commands require both an image and a drawing layer in the map.

The Pick Color command in the Tools toolbar for images is also used with tracing to specify the tracing color. To specify a tracing color, switch to the image as the active layer, use the Pick Color command to choose the color as the foreground color, click into the foreground color well and use the More... option to specify that color as a custom color. Next, switch back to the drawing and click into the Tracing Color well, click into the More... option and choose the previously saved custom color. (Future editions of Manifold will add the Pick Color command directly to the Tracing toolbar to avoid this circuitous process.)
Transform Toolbar

The Transform toolbar is loaded with operators that can accomplish "one step" editing and transformation of the contents of the active window. The transform toolbar’s operators box will be loaded with the applicable operators for whatever type of window is active (drawing, image, etc.). The transform toolbar is also used to make spatial selections using commands such as Select Touching or Select Contained within that work between sets of objects.

Because transform operators are highly specific to the type of window that is active, the transform toolbar is covered in detail within topics for each different component that has transform toolbar operators:

- Transform Toolbar - Drawings
- Transform Toolbar - Images
- Transform Toolbar - Tables

The transform toolbar uses the same general interface in all cases. We can consider the transform toolbar for drawings as an example. The transform toolbar makes changes throughout the entire drawing using the specified operators. Transform operators can create new objects, delete objects, change objects (for example, splitting them) and select objects using various algorithms such as the location of a shortest path.

Target box  Operator box  Parameter box

The Transform toolbar consists of three boxes, from left to right: a target box, the operator box, and a parameter box.

- **Target box**  Also known as the scope box. The objects that will be affected, altered or which will control the operation. Choices in the target box will be [All Objects], [Selection] or the names of any saved selections that have been saved in the Selections pane. The illustration above shows that the operator will be applied to all objects in the drawing.

- **Operator box**  The function to be applied. The operator box is context sensitive and will show only those operations that make sense for drawings. The example shows we will add border buffer zones.

- **Parameter box**  The value to be used. Depending on the operator, this may be another object set or a value entered by the user. Many operators, such as Boundaries do not require a parameter. The parameter box will not be enabled for such operators. In the example above we use 10000 for the size of the border buffer zones, the size being specified in the native units of the drawing.

We use dark blue, black, and violet bold face fonts in this topic to distinguish the three boxes to make examples and explanation more clear. In "real life," Manifold uses the same black font color in all three Transform toolbar boxes.

Using the Transform toolbar

1. Click on the map layer or drawing that contains the target objects.
2. Make a selection if the operation is to be applied just to the selection.
3. Choose the desired operator in the operator box.
4. Choose or specify a value in the parameter box, if this operation requires it.
5. Press Apply.
6. Some operators create new objects. Any new objects created will be created in the active
drawing/layer and will now be the Selection. Move the Selection to a new layer/drawing if you wish to
keep these objects organized separately from the original objects in the drawing (almost always a
good idea).

Themes and Toolbars

Themes can be used in the Query and transform toolbars. Object sets defined on themes will always use the
name of the theme's parent drawing. For example, if there are selected objects in a theme named T whose
parent is a drawing named D the selection choice will appear in the query or transform toolbar boxes as
Selection in D.

This is done so that it is immediately clear that modifying a theme will modify the parent drawing (and thus all
other themes bound to that drawing).

See Also

Transform Toolbar - Drawings
Transform Toolbar - Images
Transform Toolbar - Tables

Context Menus

Context Menus
Context menus pop open when we right click on an item in Manifold. They are called "context" menus because
what they show depends on the context of where the right click occurred.

Most controls and other items visible on-screen in Manifold have context menus. These range from the main
toolbar to tabs in maps to column heads and individual cells in tables.

See Also

Project Pane - Context Menus
Drawings - Context Menus
Labels Components - Context Menus
Maps - Context Menus
Tables - Context Menus

Project Pane

The project pane has three context menus, depending on whether we right click on an empty portion of the project
pane, whether we right click onto a folder or we right click onto a component in the project pane.

Right Click onto an empty portion of the project pane

- Paste  Paste Clipboard data as a new component.
- Paste As Like standard Windows Paste, but allows pasting into a
different type of component when feasible. For example, we might copy a geocoded table and paste it as a
drawing. Appropriate intermediate dialogs will be raised as necessary to accomplish the Paste As operation.
- Create Create a new component in the project from the standard
list of components available.

Right Click onto a folder
Delete  Delete the folder and all of its contents
Create  Create a new component within the folder from the standard list of components available in a project.
Rename  Rename the folder.
Properties  Show the properties of the folder such as the name and an editable Description field.

Right Click onto a component

Open  Open the component in a window. If it is already open, activate the opened window and move it above other open windows.
Open in New Window  Open the component in a window. If it is already open in a window, open it in a new window.
Open Data Source  Appears for components linked from an external data source. Opens the Database Console and connects to the component data source.
Run  Appears for executable components, such as scripts and queries. Run the component.
Run under Debugger  Appears for scripts when the debugger has been installed. Run the script under control of the debugger.
Cut  Copy the component to the Windows clipboard and then delete it from the project.
Copy  Copy the component to the Windows clipboard.
Paste  Enabled when the Windows clipboard has contents compatible with the project pane. Paste the contents of the Windows clipboard into the project. For example, if we copy an image in PhotoShop we can use Paste to create an image in the project pane using the contents of the Clipboard.
Paste As  Like Paste, but allows pasting into a different type of component when feasible. For example, we might copy a geocoded table and paste it as a drawing. Appropriate intermediate dialogs will be raised as necessary to accomplish the Paste As operation.
Delete  Delete the component.
Duplicate  Create a copy of the component in the project, using a similar name constructed by appending or iterating a number. For example, duplicating “Drawing 1” will result in a copy called “Drawing 2”.
Create  Create a new component in the project from the standard list of components available.
Share  Appears when Enterprise Edition is in use. Save a component into the Enterprise server. Use this command to load the Enterprise server with shared components.
Get Latest Version  Appears with Enterprise Edition. Fetches the latest version of a shared component. If we are working with a shared component and someone has checked out the component we are using and altered it, we can use Get Latest Version to fetch the latest version.
Check In  Appears with Enterprise Edition. Saves the edited version of a component back to the Enterprise server and makes the component read-only in our project. When we are
done editing a shared component we can **Check In** to save the changed version of the component to the Enterprise server. After we check in, any user getting that component will get the newly edited version.

**Check Out**

Appears with Enterprise Edition. Gets the latest version of a shared component and makes it editable exclusively by us. Only one user at a time can check out a component. Other users can still link the component, but they will not see any changes we make until we check in that component.

**Undo Check Out**

Appears with Enterprise Edition. Enabled if we have used **Check Out** to get a component for editing. Abandons any changes we have made to a shared component on our local system, gets the latest version from the server and makes it read-only.

**Cached**

Appears with Enterprise Edition. Shared components are cached by default. We can designate a component to be uncached by right clicking on a shared component in the project pane and choosing **Cached** from the pop-up context menu to toggle the check mark next to the **Cached** entry. Tool tips and the project pane status bar will show the cached/uncached status of each shared component. See the Cached and Uncached Components topic.

**Unshare**

Appears with Enterprise Edition. Convert a shared component into a local component. The data from the Enterprise server will be downloaded to the local project and all connection to the Enterprise server for this component will be ended.

**Refresh Data**

Refresh (update) the contents of linked components.

**Relink**

Relink a linked component that has been disconnected from its data source.

**Unlink**

Unlink a linked component to create a local component. Copies the data from the remote component into a local component in the project.

**Rename**

Change the name of the component. The Component Name dialog that pops up includes a checkbox to **Adjust names of dependent components**, which, if checked (the default), will automatically rename any subsidiary components. For example, if we rename a drawing called **Mexico Drawing** to **Central Mexico Drawing**, then the drawing's table that was originally called **Mexico Table** will automatically be renamed **Central Mexico Table**.

**Export**

Export the component to a file outside of Manifold.

**Print**

Print the component.

**Assign Projection**

Alter the projection assigned to a component without altering component data using the Edit - Assign Projection dialog.

**Change Projection**

Alter the projection of a component and re-project component data using the Edit - Change Projection dialog.

**Properties**

Show the properties of the component such as the name and an editable **Description** field.

**Tech Tip**

Components in the project pane may also be renamed by a “slow double-click” or by pressing the **F2** key. Press **Enter** to commit the renaming operation without renaming dependent components, or press **SHIFT-Enter** to
commit the renaming operation and to also adjust names of any dependent components as if the *Adjust names of dependent components* option in the *Component Name* dialog had been selected.

**See Also**

- Project Pane - Open Data Source
- Project Pane - Paste as Surface
- Project Pane - Paste as Drawing
Project Pane - Open Data Source
Right clicking a component that is linked from an external data source in the Project pane and then choosing Open Data Source will open the Database Console and connect to the component data source.

Opening a component linked from an external data source in its own window, and then choosing the Open Data Source command in the component menu (for example, choosing Drawing - Open Data Source when a linked drawing window is open) does the same thing.

See Also

Database Console
Project Pane - Paste as Drawing

The **Paste as Drawing** dialog appears when we copy a component and paste it as a drawing.

**Pasting Tables as Drawings**

- **Select All** - Paste all columns.
- **Select None** - Do not paste any columns except ID.
- **Select Inverse** - Do not paste all current Paste As columns and paste all other columns. A fast way to use all but one column: click **Select None**, specify a Paste As choice for the one column not desired and then click **Select Inverse**.

**X / longitude**
Choose which field in the table should be used for an X or longitude coordinate.

**Y / latitude**
Choose which field in the table should be used for a Y or latitude coordinate.

**Latitude / longitude coordinates**
Check if the values in the X / longitude and Y / latitude boxes are in degrees latitude and longitude. If this box is not checked the surface will be created in Orthographic projection treating the X and Y values as meter-based coordinates.

**Skip zero latitude / longitude records**
If checked, any records with a zero value in the latitude and longitude fields will not have an object created in the drawing. Checked by default. This is a safety measure so that when pasting geocoded tables that have zero values for latitude and longitude (and thus, are not really geocoded) spurious points will not be created for the zero valued records.
Project Pane - Paste as Surface

The Paste as Surface dialog appears when we copy a component and paste it as a surface.

**Pasting Images as Surfaces**

- **Take Z From**
  Specify the channel value that will be used for the elevation. An RGBA image, for example will offer Alpha, Red, Blue, Green or Intensity choices.

- **Minimum Z**
  What elevation the lowest pixel value corresponds to.

- **Maximum Z**
  What elevation the highest pixel value corresponds to.

- **(units box)**
  Units of measure to be used.

**Pasting Tables as Surfaces**

- **X / long**
  Choose which field in the table should be used for an X or longitude coordinate.

- **Y / lat**
  Choose which field in the table should be used for a Y or latitude coordinate.

- **Latitude / longitude coordinates**
  Check if the values in the X / long and Y / lat boxes are in degrees latitude and longitude. If this box is not checked the surface will be created in Orthographic projection treating the X and Y values as meter-based coordinates.

- **Height**
  Choose the field to be used for elevation values for the surface.

- **Type**
  The data type to use for elevation values in the surface. If the originating field is a different type, Manifold will convert to the given type when writing to the surface.

- **Margins**
  Margins in the specified unit of measure (shared with the Pixel Size combos) in X and Y directions that the surface should be extended beyond the bounding box of the objects or records being pasted to create the surface. This allows extending the created surface by some margins beyond the data being pasted to create the surface, allowing some interpolation at the edges for smoother edge effects in some cases. 0 by default.

- **Pixel Size**
  Size of pixels in the created surface, in the specified unit of measure.

- **Same size in X and Y direction**
  If checked (default), any edits in the X pixel size will also update the Y pixel size, for "square" pixels.

- **Method**
  Gravity, Kriging, Median-Polish Kriging, No Interpolation or Triangulation (flat). The type of interpolation algorithm (model) used to create the surface. Options other than Kriging will appear only if the optional Surface Tools extension has been installed.

- **Neighbors**
  Appears when using the Gravity, Kriging or Median-Polish Kriging method. The number of neighboring points to consider when making the interpolation. Manifold will automatically use all points as neighbors for interpolations up to 1000 points. Note that it is faster to use all points as neighbors up to 1000 points than it is to use some subset of points (such as, only...
990 neighbors out of 1000 points).

**Model**
Appears when using the Kriging or Median-Polish Kriging method.

*Auto* (default), Exponential, Gaussian, Linear, Power, Rational or Spherical interpolation models. The *Auto* setting allows Manifold to choose which of the interpolation models it thinks will work best in this case.

Linear, Power, Rational or Auto choices will appear only if the optional Surface Tools extension has been installed.

**Use only Voronoi neighbors**
Interpolate over each location using only those neighbors that would be adjacent in a Voronoi diagram. This helps achieve a balance between a too-smooth interpolation achieved with a high number of neighbors and a too-coarse interpolation achieved with a low number of neighbors. Using this option will increase processing time.

**Use radius**
Appears when using the Gravity, Kriging or Median-Polish Kriging method. The distance in the given units from any data point over which an interpolation can extend. Not checked by default to allow Manifold to choose the radius.

**Save error surface as**
Appears when using the Kriging or Median-Polish Kriging method. Allows saving of an error surface using the given numeric type.

**Set corner values to**
Appears when using the triangulation method. Optionally set the values in the corners of the computed surface to the given values. If this option is used, the entire surface (to the full rectangular extents) will be interpolated using the corner values as anchors. If this option is not used, those regions outside the convex hull of the existing data points will remain invisible.

**Pasting Drawings as Surfaces**

Pasting a drawing as a surface is similar to pasting a table as a surface except that there is no need to specify the fields to be used as latitude and longitude, since the coordinates of objects in the drawing are built into the geometry of the drawing.

Likewise, there is no need to specify whether the coordinates are latitude and longitude coordinates, since the coordinate system used by the drawing is also known. When pasting a drawing as a surface, the surface will inherit all projection parameters from the drawing.

When pasting a drawing as a surface, Manifold can use point and line objects to create a surface. Point objects are taken as point data from which the surface is interpolated. Line objects will have point data created at each coordinate ("inflection point") that defines the line.

**Height**
Choose the field to be used for elevation values for the surface.

**Type**
The data type to use for elevation values in the surface. If the originating field is a different type, Manifold will convert to the given type when writing to the surface.

**Margins**
Margins in the specified unit of measure (shared with the Pixel Size combos) in X and Y directions that the
- Surface should be extended beyond the bounding box of the objects or records being pasted to create the surface. This allows extending the created surface by some margins beyond the data being pasted to create the surface, allowing some interpolation at the edges for smoother edge effects in some cases. 0 by default.

**Pixel Size**  
Size of pixels in the created surface, in the specified unit of measure.

**Same size in X and Y direction**  
If checked (default), any edits in the X pixel size will also update the Y pixel size, for "square" pixels.

**Method**  
Gravity, Kriging, Median-Polish Kriging, No Interpolation or Triangulation (flat). The type of interpolation algorithm (model) used to create the surface. Options other than Kriging will appear only if the optional Surface Tools extension has been installed.

**Neighbors**  
Appears when using the Gravity, Kriging or Median-Polish Kriging method. The number of neighboring points to consider when making the interpolation. Manifold will automatically use all points as neighbors for interpolations up to 1000 points. Note that it is faster to use all points as neighbors up to 1000 points than it is to use some subset of points (such as, only 990 neighbors out of 1000 points).

**Model**  
Appears when using the Kriging or Median-Polish Kriging method. Choose Auto (default), Exponential, Gaussian, Linear, Power, Rational or Spherical interpolation models. The Auto setting allows Manifold to choose which of the interpolation models it thinks will work best in this case. Linear, Power, Rational or Auto choices will appear only if the optional Surface Tools extension has been installed.

**Use only Voronoi neighbors**  
Interpolate over each location using only those neighbors that would be adjacent in a Voronoi diagram. This helps achieve a balance between a too-smooth interpolation achieved with a high number of neighbors and a too-coarse interpolation achieved with a low number of neighbors. Using this option will increase processing time.

**Use radius**  
Appears when using the Gravity, Kriging or Median-Polish Kriging method. The distance in the given units from any data point over which an interpolation can extend. Not checked by default to allow Manifold to choose the radius.

**Save error surface as**  
Appears when using the Kriging or Median-Polish Kriging method. Allows saving of an error surface using the given numeric type.

**Set corner values to**  
Appears when using the triangulation method. Optionally set the values in the corners of the computed surface to the given values. If this option is used, the entire surface (to the full rectangular extents) will be interpolated using the corner values as anchors. If this option is not used, those regions outside the convex hull of the existing data points will remain invisible.

---

**Notes**
Images that represent raster data such as elevations, temperatures and other factors are often visualized as surfaces.

If a field named z, elevation, elev, e, height, hgt or h (case insensitive) exists in the drawing or table, Manifold will offer that by default as the field in the Height box. Files named longitude, latitude, lon, long, lat, x or y (case insensitive) will be offered as the X / long or Y / lat box fields by default.

Attempting to create a surface larger than 16 million pixels (4,000 x 4,000) will trigger a confirmation dialog that will ask "Surface size is (x dimension) x (y dimension) pixels. Paste anyway?"

The Paste as Surface dialog will automatically compute and display the dimensions and size of the surface that will be created. Surface sizes larger than gigabytes are displayed in terabytes ("Tbytes"), petabytes ("Pbytes") and exabytes ("Ebytes"), although as a practical matter on most computers such very large surfaces are too large to create or display due to a lack of free disk space or other limitations.

Using the No interpolation option when pasting a surface from a drawing or a table will result in a surface being created as a series of facets that are not interpolated between the exact location of the given data points. Rather than appearing smooth when zoomed in, the surface will appear pixilated.

After pasting a drawing or table as a surface, when the surface is opened the Notes pane will list the interpolation parameters that were used to create the surface.

See Also

See the Creating Surfaces from Drawings and Tables topic for some notes and an example.

Drawings

Drawings - Context Menus

The drawing window has only one context menu, which appears when we right click onto any object.

Right Click onto an Object

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cut</td>
<td>Windows clipboard cut operation. Copies object to the clipboard and delete it from the drawing.</td>
</tr>
<tr>
<td>Copy</td>
<td>Windows clipboard copy operation. Copies object onto the clipboard without deleting it.</td>
</tr>
<tr>
<td>Paste</td>
<td>Paste the contents of the Windows clipboard into the drawing. Creates new objects in the drawing, replacing any objects that were selected. New objects will be selected.</td>
</tr>
<tr>
<td>Paste Append</td>
<td>Like Paste, but does not delete and replace any objects that were previously selected.</td>
</tr>
<tr>
<td>Delete</td>
<td>Permanently delete this object.</td>
</tr>
<tr>
<td>Duplicate</td>
<td>Duplicate the object. It is wise to immediately move the new object while it is selected. This avoids creating coincident objects that might later cause confusion.</td>
</tr>
<tr>
<td>Orthogonalize</td>
<td>Move coordinates to orthogonal grid positions. Used to trim insignificant digits.</td>
</tr>
<tr>
<td>Segmentize</td>
<td>Add redundant coordinates to increase the number of coordinates used to define the object. See the Segmentization topic for discussion.</td>
</tr>
<tr>
<td>Simplify</td>
<td>Remove coordinates that define the object to simplify the shape of the object.</td>
</tr>
</tbody>
</table>
Center  Pan the view to center this object in the window.
Zoom  Zoom to fit this object.
Coordinates  Launches the Object Coordinates dialog to allow direct editing of the coordinates that comprise the object.
Fields  Launches the Object Fields dialog that shows any table fields associated with this object.

The **Cut**, **Copy** and **Delete** functions within the context menu of a drawing work slightly different than their equivalents found in the **Edit** menu. The **Edit** menu versions work with the selection. The context menu versions of these commands work on those that object that has been right clicked.

**Example**

Right click onto an area in the drawing and choose **Delete** from the context menu. The area will be deleted.

**Right Click onto an Object that has been Selected for Editing**

**CTRL-ALT** click on an object to select it for editing. Editing handles will appear at the coordinates that define the object. A right click on the object (either on a coordinate of the object or on a line segment between coordinates) calls up the following menu:

- **Cut**  Windows clipboard cut operation. Copies object to the clipboard and delete it from the drawing.
- **Copy**  Windows clipboard copy operation. Copies object onto the clipboard without deleting it.
- **Paste**  Paste the contents of the Windows clipboard into the drawing. Creates new objects in the drawing, replacing any objects that were selected. New objects will be selected.
- **Paste Append**  Like **Paste**, but does not delete and replace any objects that were previously selected.
- **Delete**  Permanently delete this object.
- **Branch**
  - **Delete**  - Delete this branch (not enabled if there is only one branch - use the regular **Delete** choice.)
  - **Duplicate**  - Duplicate this branch. Use with care, as it is easy to lose track of multiple branches if they are created on top of one another.
  - **Split**  - Enabled if clicked on a coordinate. Split the object into two branches at this coordinate.
- **Segment**
  - **Delete**  - Delete this segment. It is not possible to delete one of the last two segments of a line branch or one of the last three segments of an area branch.
  - **Delete / Split**  - Delete this segment splitting the branch it is in into two branches. Only available when editing a line object. It is not possible to split the branch by deleting its first or last segment.
- **Coordinate**
  - **Add**  - Add a new coordinate at the location clicked. Such a click will not likely be exactly on the line segment, as a click reasonably near a selected object will be interpreted as involving that line even if the click was not exactly on the line segment.
  - **Add on Segment**  - Add a new coordinate to the object exactly on the line segment nearest to the click.
  - **Add Mid-Segment**  - Add a new coordinate to the object exactly in the middle of the line segment nearest to the
click.

**Delete** - Delete this coordinate. It is not possible to delete one of the last two coordinates defining a branch of a line or one of the last three coordinates defining a branch of an area.

**Delete / Split Branch** - Delete this coordinate and split this branch into two branches at that location. Used with lines.

**Duplicate** - Duplicate this coordinate. Used with lines and areas. It is wise to immediately move the new coordinate / edit handle to avoid confusion that might be caused by coincident (redundant) coordinates.

**Duplicate**

**Orthogonalize** Move coordinates to orthogonal grid positions. Used to trim insignificant digits.

**Segmentize** Add redundant coordinates to increase the number of coordinates used to define the object. See the Segmentization topic for discussion.

**Simplify** Remove coordinates that define the object to simplify the shape of the object.

**Center** Pan the view to center this object in the window.

**Zoom** Zoom to fit this object.

**Coordinates** Launches the Object Coordinates dialog to allow direct editing of the coordinates that comprise the object.

**Fields** Launches the Object Fields dialog that shows any table fields associated with this object.

The **Cut**, **Copy** and **Delete** functions within the context menu of a drawing work slightly different than their equivalents found in the **Edit** menu. The **Edit** menu versions work with the selection. The context menu versions of these commands work on those that object that has been right clicked.
Drawings - Object Coordinates

The Object Coordinates dialog appears as the Coordinates choice in the context menu when we right click on an object in a drawing. The dialog allows direct editing of the coordinates that comprise the object.

Please read the Coordinates topic if you have not already done so. This topic describes how objects are defined by the coordinates that are used to draw them in a "connect the dots" fashion.

The Object Coordinates dialog displays the list of coordinates that define the object that was right clicked. Editing this list will change the shape and/or position of the object. A + sign in the first column indicates the beginning of each branch of an object.

- **New** - Create a new coordinate in the list that is a duplicate of the currently selected coordinate (lines and areas only).
- **New Branch** - Insert a new branch (lines and areas only). A branch is a separator between coordinate lists used to define complex objects such as areas that contain holes.
- **Break Branch** - Break branch at highlighted coordinate (lines only, disabled unless there are at least two points before and after highlighted coordinate).
- **Delete** - Removes highlighted coordinate (lines and areas only, disabled for area branches containing three points and line branches containing two points).
- **Delete Branch** - Delete highlighted branch (lines and areas only, disabled when there is only one branch left).
- **Move to Top** - Move highlighted coordinate to start of branch (lines and areas only, disabled when coordinate is already at start of branch).
- **Move Up** - Move highlighted coordinate one step up (lines and areas only, disabled when coordinate is already at start of branch).
- **Move Down** - Move highlighted coordinate one step down (lines and areas only, disabled when coordinate is already at end of branch).
- **Move to Bottom** - Move highlighted coordinate to end of branch (lines and areas only, disabled when coordinate is already at end of branch).
Native Coordinates - Switch between coordinates within the projection coordinate system native to the drawing and Latitude and Longitude values.

Copy - Copy the list of coordinates to the Windows Clipboard.

Print - Print the list of coordinates through the standard print dialog.

This dialog is most frequently used to adjust the position of a point. Because a point is defined by only one coordinate it is easy to open this dialog and to change the X / Longitude and Y / Latitude values for the point. The dialog is less frequently used to edit areas and lines since such objects in a geographic context may be defined by thousands of coordinates.

One usage for areas or lines is when we wish to draw simple areas and lines in exact positions on the map. For example, suppose we want to draw a rectangular area within a Latitude / Longitude projection map that extends from latitude 38 to latitude 39 and from longitude 100 to longitude 101. An easy way of doing this is to draw a rectangle that approximately covers this region using the Insert Box tool from the Tools toolbar. We can then right click on this rectangle, choose Coordinates and then use the Object Coordinates dialog to change the four coordinates that define the rectangle into the exact values we want for the corners of the rectangle.

Native Coordinates

Native coordinates are numbers stored within the drawing that Manifold uses internally as positional coordinates. These numbers are shown in the Object Coordinates dialog. They are also used in the X (I) and Y (I) intrinsic field columns.

Projected coordinates are native coordinates adjusted with the local scale and local offset parameters of the coordinate system (projection) in use. In many cases, projected coordinates are the same as native coordinates (because of the local scale and offset values) but sometimes they are different.

Tech Tip

The Object Coordinates dialog may be invoked by first selecting an object for editing using smart mouse. Right clicking on one of the edit handles and then choosing Coordinates from the context menu will launch the Object Coordinates dialog with that edit handle's coordinate already highlighted and in view. See the Editing Objects topic.

Notes

Technically speaking, each line in the list defining the object is a coordinate pair. It is Manifold jargon to use the word "coordinate" to mean the pair of X / Longitude and Y / Latitude coordinates.

It is not possible to use this dialog to delete the last three coordinates in an area, the last two coordinates in a line or the last coordinate defining a point. To delete objects, select them and use Edit - Delete.

In the example noted above, of creating a rectangle covering one degree of longitude and latitude, if we intend to use this rectangle in projected maps we should Segmentize it after editing it to our satisfaction in the Object Coordinates dialog.

See Also

View - Structure
Editing Objects
Drawings - Object Fields

The **Object Fields** dialog appears as the **Fields** choice in the context menu when we right click on an object in a drawing. The dialog shows any data fields for this object in tables associated with the drawing.

The **Object Fields** dialog will show all fields in the drawing’s table. In addition, it will show the system-generated intrinsic fields for that object if desired. The status bar at the bottom of the dialog will report how many fields are displayed out of all available.

Although the dialog will remember previous settings, the **Object Fields** dialog tries to make sure it shows at least some fields on startup, even if the various Show ... filters have been set so that all fields are hidden. First the Show Blanks filter will be turned on to show fields with blank values, then the Show Intrinsics mode and finally Show Read-Only fields.

The **Object Fields** dialog is called with pre-loaded values when Instant Data mode is on and a new object is created.

**Note:** if fields for more than one object are to be reviewed, the Info Pane is usually more efficient than using the **Object Fields** dialog.

**Tech Tip: Speed on Opening**

The first time the **Object Fields** dialog is opened it may open more slowly than when subsequently opened. The reason is that some intrinsic columns are computed the first time the dialog is opened but their values may be fetched from cache during subsequent opens.
Some intrinsic columns, for example, Area (I), take a long time to compute, so Manifold caches their values whenever possible. The first time we access such a column (say, by invoking the Object Fields dialog, which queries the value of each intrinsic column, even if the intrinsic columns are initially hidden), Manifold will do the necessary computations and will save the resulting values into cache. On subsequent accesses Manifold will not need to compute the intrinsic column but can simply use values already in cache.

**Tech Tip: Field Filtering**

The Object Fields dialog filters the columns it shows, choosing which fields are displayed based on the settings of the Show Blanks, Show Intrinsics and the Show Read-Only buttons. By default, the dialog does not show the intrinsic and read-only columns. Any changes made to the filter settings are saved within Manifold so double-clicking an object, changing filter settings, closing the dialog and double-clicking another object will use the filter settings saved from the previous time the dialog was launched.

An exception to this rule happen if the Object Fields dialog starts up with all columns hidden due to filter settings. In that case Manifold will alter filter settings to show at least some columns, since opening the dialog without any columns visible might confuse inexperienced users.

When choosing what columns to show in such cases, Manifold will first turn on display of columns with blank values. If no columns become visible, the system will turn on display of read-only columns and then finally, if necessary, intrinsic columns. Note that one result of this strategy to help new users is that double-clicking an object in a drawing linked from a table or a query will by default display non-intrinsic columns.

**See Also**

- Instant Data
- Intrinsic Fields in Tables
- Info Pane

**Labels Components**

**Labels Components - Context Menus**

The labels component window has only one context menu, which appears when we right click onto any label.

**Right Click onto a Label**

- **Cut**
  Windows clipboard cut operation. Copies label to the clipboard and delete it from the labels component.

- **Copy**
  Windows clipboard copy operation. Copies label to the clipboard without deleting the label.

- **Paste**
  Paste the contents of the Windows clipboard into the labels component. Pastes labels in the labels component, replacing any labels that were selected. New labels will be selected.

- **Paste Append**
  Like Paste, but does not delete and replace any labels that were previously selected.

- **Delete**
  Permanently delete this label.

- **Edit**
  Edit label text.

- **Duplicate**
  Create a duplicate of this label in the same location and select it, replacing the selection. When using Duplicate it is a good idea to immediately move the duplicate label to some other position to avoid creating coincident labels.

- **Orthogonalize**
  Move coordinates to orthogonal grid positions. Used to trim insignificant digits.

- **Segmentize**
  Add redundant coordinates to increase the number of coordinates used to define the label. Enabled only for line labels. See the Segmentization topic for discussion.
Simplify

Remove coordinates that define the object to simplify the shape of the object. Enabled only for line labels.

Center

Center this label in the map.

Coordinates

Launches the Object Coordinates dialog to allow direct editing of the coordinates that comprise the label.

The Cut, Copy and Delete functions within the context menu of a labels component work slightly different than their equivalents found in the Edit menu. The Edit menu versions work with the selection. The context menu versions of these commands work on those that label that has been right clicked.

Right Click onto a Label that has been Selected for Editing

Right clicking on a label that has been selected for editing calls a context menu that contains editing commands. When right clicking on an editing handle on a label selected for editing, or onto a line segment between editing handles, the context menu will also include additional editing commands.

Duplicate

Duplicate the label. It is wise to immediately move the new label while it is selected. This avoids creating coincident labels that might later cause confusion.

Orthogonalize

Move coordinates to orthogonal grid positions. Used to trim insignificant digits.

Segmentize

Add redundant coordinates to increase the number of coordinates used to define the label. Enabled only for line labels. See the Segmentization topic for discussion.

Simplify

Remove coordinates that define the label to simplify the shape of the label. Enabled only for line labels.

Center

Pan the view to center this label in the window.

Coordinates

Launches the Object Coordinates dialog to allow direct editing of the coordinates that comprise the label.

Branch

Delete

Delete this branch. Not enabled unless there is more than one branch.

Duplicate

Duplicate this branch at exactly the same position. Use CTRL-SHIFT to drag to a different position the new branch that is thus created (which otherwise exactly overlies the previous branch).

Split

Split this branch into two branches at the clicked position.

Segment

Delete

Delete the line segment clicked along with the edit handle coordinates to either side of the segment. Not enabled if there is only one segment left.

Delete / Split Branch

Delete the line segment clicked along with the edit handles to the left and right and also split this branch into two branches at that location.

Coordinate

(Right clicking onto an edit handle)
Delete
Delete this coordinate. It is not possible to delete the last two coordinates defining a line label.

Delete / Split Branch
Delete this coordinate and also split this branch into two branches at that location.

Duplicate
Duplicate this coordinate. It is wise to immediately move the new coordinate/edit handle to avoid confusion that might be caused by coincident (redundant) coordinates.

Coordinate
(Right clicking onto a segment between edit handles)

Add
Add a new edit handle (coordinate) at the location clicked.

Add on Segment
Add a new edit handle (coordinate) on the existing line near the location clicked.

Add Mid-Segment
Add a new edit handle (coordinate) at the middle of the segment nearest to the location clicked.

Example
Right click onto an area in the labels component and choose Delete from the context menu. The area will be deleted.

Maps
Maps - Context Menus
The main context menu for maps appears when right clicking onto a layer tab. The drawing context menu appears when clicking onto an object in a drawing layer in a map.

Context Menu for Map Layer Tabs

Visible
Show or hide this layer.

Add
Add a new blank drawing, image or labels component using the map's projection, or launch the Layers dialog to add components from the project pane. The component will be added as a layer just above the active layer tab.

Cut
Windows clipboard cut operation. Copy objects or pixels to the clipboard and delete them from the drawing.

Copy
Windows clipboard copy operation. Copy objects or pixels from this layer onto the clipboard.

Paste
Paste the contents of the Windows clipboard into this layer replacing any selected items.

Paste Append
Paste the contents of the Windows clipboard into this layer.

Delete from Map
Delete this layer from the map. The component will continue to exist in the project pane and in any other maps in which it participates.

Duplicate
Make a copy of this component in the project pane and insert it in the map.

Open
Open this component in its own window. If it is already open in its own window, activate that window.
**Open in New Window**
Open this component in its own window. If it is already open in its own window, open it again in yet another window.

**Rename**
Rename this component.

**Order**
Move layer to top, up, down or to bottom of layer stack.

**Opacity**
Set the opacity of this layer. See the Layer Opacity topic and the note below regarding the Opacity dialog.

**Restrictions**
Specify if this layer allows clicking, editing or selecting using mouse commands. See the Layers pane topic for more information on layer restrictions.

**Center**
Center the active layer in the map window.

**Zoom**
Zoom to the active layer in the map window.

**Zoom to Selection**
Zoom to fit the selection in the active layer in the map window.

**Match**
Enabled for image or surface layers. Crop / expand / resample / re-project the active layer so that matches the image or surface specified. Often used to prepare an image so that it is the same size and coordinate system as an another for use as a mask or other purposes.

**Register**
Move and re-scale this drawing to specify its geographic location. A manual operation that is rarely used. See the Georegistration topic for easier methods.

**Project to Map**
Re-project this layer’s component to match the projection currently used in the map. This is a permanent alteration in the data. When re-projecting an image or a surface, this command will automatically use equal values for the X and Y local scale parameters, thus resulting in geographically "square" pixels.

**Transfer Heights**
Transfer height values from a surface to a drawing. Available only if the Surface Tools extension has been installed.

**Transfer Selection**
Transfer selections between components.

**Visible Area**
Compute area visible from one or more locations in a surface. Available only if the Surface Tools extension has been installed.

**Use Projection**
Apply the native projection of this component as the projection used by the map.

**Properties**
View the properties dialog for this layer.

---

**Right Click onto an Object (Drawing Layers)**

**Cut**
Windows clipboard cut operation. Copies object to the clipboard and delete it from the drawing.

**Copy**
Windows clipboard copy operation. Copies object onto the clipboard without deleting it.

**Paste**
Paste the contents of the Windows clipboard into the drawing. Creates new objects in the drawing, replacing any objects that were selected. New objects will be selected.

**Paste Append**
Like Paste, but does not delete and replace any objects that were previously selected.
Delete
Permanently delete this object.

Duplicate
Duplicate the object. It is wise to immediately move the new object while it is selected. This avoids creating coincident objects that might later cause confusion.

Orthogonalize
Move coordinates to orthogonal grid positions. Used to trim insignificant digits.

Segmentize
Add redundant coordinates to increase the number of coordinates used to define the object. See the Segmentization topic for discussion.

Simplify
Remove coordinates that define the object to simplify the shape of the object.

Center
Pan the view to center this object in the window.

Zoom
Zoom to fit this object.

Coordinates
Launches the Object Coordinates dialog to allow direct editing of the coordinates that comprise the object.

Fields
Launches the Object Fields dialog that shows any table fields associated with this object.

Opacity Dialog
An alternative to setting layer opacity via the % edit box in the Layers pane is to use the opacity dialog that pops up from the context menu for map layer tabs. Right click onto a layer tab and choose Opacity to open the dialog:

The opacity dialog allows interactive setting of transparency by moving a slider bar. Check the Preview box to see the effect in real time. We can also click into the edit box and change the opacity value number.

Nomenclature

Transparency and opacity are two terms that mean the same concept viewed from different directions. When something is completely opaque it is not at all transparent. When something is perfectly transparent it may be said to have zero percent opacity.

Which word is used depends on the discussion. When imagining layers stacked up above each other like transparent sheets it is conceptually clearer to use the word transparency. When discussing a specific percentage of light transmission to be applied via a slider bar in a dialog most applications use the word opacity.

The convention in the graphics arts editing software industry is to adjust layer opacity with controls that set a number from 0% to 100% opacity, so that an image with 100% will be fully opaque and not allow any view of an image underneath it. Manifold follows this convention. This convention persists in the graphics arts industry even though the technical implementation of transparency effects is done using an alpha channel within RGBA images where the higher the value of the alpha channel (from 0 to 255) the higher the transparency.

One therefore encounters the slight conceptual dissonance of increasing opacity with higher numbers (up to 100%) in dialogs and other user interfaces while the internal data sets use numbers (alpha channel values) in which opacity decreases with higher numbers. Since we rarely set alpha values by hand this is not so bad. Alpha values are normally set using various tools, such as erasers, or masks. In the case of masks, the darker the mask region the lower the alpha value is and thus the higher the opacity. From a casual conceptual view this is very acceptable because it leads to an effect where black regions of masks cause full opacity and white regions...
of masks cause full transparency. Since we are used to thinking of "white space" as being transparent this works well as a natural mnemonic for the effects of masks.
Maps - Register

The **Register** dialog appears when choosing **Register** from a layer tab context menu. Right click on a layer tab in a map and then choose **Register** to open this dialog.

The **Register** dialog allows us to move and rescale this drawing to specify its geographic location. It is a manual operation that is rarely used. On occasion, after using automated georegistration the **Registration** dialog will be used to "nudge" an image into slightly different alignment to suite the user's taste.

See the Georegistration topic for easier methods of georegistering images and drawings. See the Manual Georegistration topic for discussion of manual georegistration issues, including the use of the **Register** dialog.

---

**Register** dialog:

- **Local offset**: Shifted version of the external coordinate system peculiar to this component. Specifies shift in X (longitude) and Y (latitude) direction in the given units.
- **Local scale**: Scaled version of the external coordinate system peculiar to this component. Specifies scale factor to be applied to external coordinate system.
- **Preview**: Show effect of changes in the map window.

**Units box**: Choose the unit of measure. All "unprojected" maps are in Degrees, Arc Minutes or Arc Seconds. Most projected maps are in meters.

Changing the **Local scale** settings changes the scale of the component to make it bigger or smaller. Changing the **Local offset** moves the component horizontally or vertically. With great patience and conceptual understanding it is possible to use the **Register** dialog to move and re-scale overhead view images so they are aligned to drawings in Orthographic projection. A much better idea is to use Georegistration tools to do the job automatically.

The **Register** dialog is used occasionally to create special effects or other expert purposes. The following example shows one possible usage.

**Example: Create a "Drop Shadow"**

We can give our drawing layers a sense of three dimensionality using the following procedure.

**Step 1: Prepare a Map of Mexico**

- Import a drawing of Mexico. Apply a thematic format.
- Right click on the drawing in the project pane and choose **Copy**.
- Right click again in the project pane and choose **Paste As - Drawing**. This creates an exact copy of the drawing.
- Format the copy so it is black in both foreground and background colors.
- Create a map using **mexico** and the copy. Open the map.
The illustration shows the map window with the `mexico` layer above the copy.

If we double click on the `mexico` layer to turn it off we can see the copy underneath.

**Step 2: Use the Register Dialog**
Right click on the copy layer (the all-black copy of \textit{mexico}) and choose \texttt{Register}. Check the \texttt{Preview} box so we can see the action of the dialog. Click on the up spin button on \texttt{Local offset X}. This moves the \texttt{CopyOfmexico} layer to the right.
Click on the up spin button on **Local offset Y**. This moves the **CopyOfmexico** layer down. The result is what appears to be a shadow below the **mexico** layer. When we are happy with the effect we click **OK**.

**Variations**

The effect shown above is exaggerated. Smaller offsets in **Local offset** usually look better.
Combine this technique with other effects. The illustration above was created by making two more copies of the \textit{mexico} drawing and in both of them deleting all objects except the highlighted provinces. We see a map with four layers: the topmost two layers contain the two provinces with a "drop shadow." Below them is the \textit{mexico} layer at a 60\% Layer Opacity value.

\textbf{See Also}

\texttt{Edit - Change Projection} for more information on using offsets and their relationship to other projection parameters such as false easting and false northing.

\textbf{Tables}

\textbf{Tables - Context Menus}

Tables have several different context menus that appear. Right click onto column heads, a cell, or a record handle to see the relevant context menu.

\textbf{Context Menu for Column Heads}

\begin{verbatim}
<table>
<thead>
<tr>
<th>Company Name</th>
<th>Company Name</th>
<th>Company Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alfreds Furtikste</td>
<td>Ana Trujillo Emareda</td>
<td>Arto</td>
</tr>
<tr>
<td>Arto</td>
<td>Around the Horn</td>
<td>Berglunds snakklopp</td>
</tr>
<tr>
<td>Berglunds snakklopp</td>
<td>Around the Horn</td>
<td>Berglunds snakklopp</td>
</tr>
<tr>
<td>Chris</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
\end{verbatim}

A \textit{column} is a field in the database table. In the notes that follow, "column" is a synonym for "field." The context menu for columns appears when right clicking onto a column head.

\begin{itemize}
  \item \textbf{Add - Column} Add a new column to this table.
  \item \textbf{Add - Active Column} Add a column in which values are determined by the results of a script. See \texttt{Active Columns}.
\end{itemize}
| **Add - Rank Column** | Add a column used to rank records by the results of a Decision Support System query. See Rank Columns.
| **Edit** | Enabled for Active Columns and Rank Columns. Adjust the source of this column (script or criteria).
| **Edit Script** | Enabled for Active Columns. Edit the script associated with this column.
| **Flatten** | Enabled for Active Columns, Rank Columns and Relations. Convert the column to a regular column by using the current values in the cells.
| **Flatten All** | Convert all Active Columns, Rank Columns and Relations to regular columns by using the current values in the cells.
| **Recompute** | Recompute the values in an Active Column.
| **Cut** | Windows clipboard cut operation. Copy selected records to the clipboard and delete them from the table.
| **Copy** | Windows clipboard copy operation. Copy selected records from this table onto the clipboard.
| **Paste** | Paste the contents of the Windows clipboard into this table, replacing selected records.
| **Paste Append** | Paste the contents of the Windows clipboard into this table, appending them to the table.
| **Delete** | Permanently delete selected records from this table.
| **Rename** | Change the name of this column.
| **Transfer Rules** | Specify how the values in this column should be transferred in operations such those in the Spatial Overlay dialog.
| **Format** | Specify how values should appear in this column.
| **Width** | Specify the width of this column in pixels.
| **Hide** | Show or hide this column. To find hidden columns, use the View - Columns command in the main menu for tables.
| **Change Type** | Change the field type of this column. Note that not all field types may be freely converted. For example, the number 132 can be converted into a text string containing the characters "132", but the text value "Madrid" cannot be converted into a number.
| **Language** | Specify Windows code page to be used for this column. Different columns in the same table can use different code pages to support different languages. Enabled for ANSI text fields only.
| **Order** | Move the column to the top (leftmost position), up, down, or to the bottom (rightmost position) in the display.
| **Best Fit** | Arrange the width of this column so that the values in the database will fit in it without extra space.
| **Best Fit All** | Arrange the width of all columns so that the values in the database will fit in them without extra space.
| **Best Fit Titles** | Arrange the width of all columns so that the full text of all column head titles is visible.
Identity  Select the ID column to be used in tables created with queries that have more than one identity column. See the Queries Using Multiple Tables topic. Enabled in query tables.

Find  Find a particular value in this column. Regular expressions may be used to search for patterns.

Replace  Find and replace values in this column. Regular expressions may be used to search for patterns and to specify their replacements.

Context Menu for Cells

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cut</td>
<td>Windows clipboard cut operation. Copy this cell's value to the clipboard and delete it from the cell.</td>
</tr>
<tr>
<td>Copy</td>
<td>Windows clipboard copy operation. Copy this cell's value to the clipboard.</td>
</tr>
<tr>
<td>Paste</td>
<td>Paste the contents of the Windows clipboard into this cell, replacing the value it contains.</td>
</tr>
<tr>
<td>Paste Append</td>
<td>Paste the contents of the Windows clipboard into this cell, appending to the value in the cell.</td>
</tr>
<tr>
<td>Delete</td>
<td>Permanently delete the contents of this cell.</td>
</tr>
<tr>
<td>More Like</td>
<td>Open a dialog allowing optional controls over a &quot;more like this&quot; search that specifies how each column should be reckoned to find additional records that are &quot;like&quot; this one.</td>
</tr>
<tr>
<td>More Like This Cell</td>
<td>Automatically find and select records that have values &quot;like&quot; this cell in this column.</td>
</tr>
<tr>
<td>More Like This Cell (Sorted)</td>
<td>Automatically find and select records that have values &quot;like&quot; this cell in this column. Sort the table so that records that are most like the value in this cell are at the top of the table.</td>
</tr>
</tbody>
</table>

A cell is any row/column intersection in a table. It is the value of a particular column for a specific record. The context menu for cells appears when right clicking onto a cell.

Context Menu when Editing Cells

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALFKI</td>
<td>Alfreds Puterkste Maric</td>
</tr>
<tr>
<td>ANATR</td>
<td>Ana Trujillo Empareda... Ana'</td>
</tr>
<tr>
<td>ANTON</td>
<td>Antonio Moreno Tapia... Arto</td>
</tr>
<tr>
<td>AROUT</td>
<td>Around the Horn</td>
</tr>
<tr>
<td>BERGS</td>
<td>Berglund's Snabblöp Chris</td>
</tr>
<tr>
<td>BLAUS</td>
<td>Player See Cell stereotype User</td>
</tr>
</tbody>
</table>
Double-clicking into a cell opens it for editing. The usual Windows context menu selections for editing values will appear when right clicking into the cell being edited.

- **Undo**: Undo the last edit operation. Only possible until the cell is closed for editing with an ENTER or by clicking on a different cell.
- **Cut**: Windows clipboard cut operation. Copy the highlighted part of the value to the clipboard and delete it from the cell.
- **Copy**: Windows clipboard copy operation. Copy the highlighted part of the value to the clipboard.
- **Paste**: Paste the contents of the Windows clipboard into this cell, replacing any highlighted part of the value.
- **Delete**: Permanently delete the highlighted portion of the value.
- **Select All**: Highlight all of the cell's value.

### Context Menu for Records

<table>
<thead>
<tr>
<th>Customer ID</th>
<th>Company Name</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALFKT</td>
<td>Alfreds Futterkiste</td>
<td>Maria</td>
</tr>
<tr>
<td>ANATR</td>
<td>Ana Trujillo Empareda</td>
<td>Ana'</td>
</tr>
<tr>
<td>ANTON</td>
<td>Antonio Moreno Taquero</td>
<td>Arto</td>
</tr>
<tr>
<td>AFOUT</td>
<td>Around the Horn</td>
<td>Thord</td>
</tr>
<tr>
<td>BERGS</td>
<td>Berglund's snabblöp</td>
<td>Chris</td>
</tr>
<tr>
<td>FHANS</td>
<td>Birger Saas Dalskrog</td>
<td>Hans</td>
</tr>
</tbody>
</table>

A **record** is any row in a table. In tables associated with drawings, each record represents the data fields for one object (a point, line or area) in the drawing. The context menu for records appears when right clicking onto a record's handle in the left border of the table.

- **Cut**: Windows clipboard cut operation. Copy this record's values to the clipboard and delete it from the table.
- **Copy**: Windows clipboard copy operation. Copy this record's values to the clipboard.
- **Paste**: Paste the contents of the Windows clipboard into this record, replacing the values it contains.
- **Delete**: Permanently delete this record.
- **More Like**: Open a dialog allowing optional controls over a "more like this" search that specifies how each column should be reckoned to find additional records that are "like" this one.
- **More Like This Record**: Automatically find and select records that have values "like" this record in this column, using those columns that are displayed in the table.
- **More Like This Record (Sorted)**: Automatically find and select records that have values "like" this record in this column. Sort the table so that records that are most like this record are at the top of the table.
- **Coordinates**: In tables associated with drawings, show the coordinates that define the drawing object for this record.
See the More Like This topic for information on the More Like This commands and dialog.
SQL Reference Guide

SQL in Manifold System

Manifold System includes a full implementation of the database language SQL. SQL is a language used in querying, updating, and managing relational databases. SQL can be used to retrieve, sort, and filter specific data to be extracted from databases. The system includes Data Definition Language (DDL) and Data Manipulation Language (DML) components that are used to manipulate the structure of the database and the data it contains. The DDL and DML capabilities are often lumped together in the term “SQL.”

Manifold System's SQL is a new implementation of SQL created by the Manifold engineering team that is optimized for spatial operations within Manifold projects. For maximum compatibility within Microsoft operating systems, Manifold SQL is designed so that it can operate as a superset of Microsoft SQL as it occurs within Microsoft's "Jet" database engine. Manifold SQL can also operate in ANSI compatibility mode.

In addition to duplicating the behavior of Jet SQL or ANSI SQL, Manifold SQL provides numerous extensions, profoundly enhanced speed in spatial usage and much greater flexibility. For connection to external databases, Manifold System uses standard Microsoft interfaces, including OLE DB, ODBC and Jet, which then allows usage of the external, linked data within projects including operations with Manifold SQL. See the Comparison with Jet and Comparison with ANSI SQL topics for a list of similarities and differences.

At times it may be convenient to work with external databases without importing their tables or linking them into a Manifold System project. For example, we might wish to alter the database or to use SQL to create a subset of data that may then be imported or linked into Manifold System. In earlier times, one would have to fire up Access or Manifold Database Commander or some other database tool to pound the external database into the desired shape. With Manifold System, one may perform such external tasks from within Manifold with no need for any external tools.

Manifold SQL is a full-power SQL implemented with the very latest in computing technology. SQL is an immense language that embeds the collective experience of the finest minds in database languages over many years. If you are new to SQL, get a good introductory book on this industry standard language.

To find specific words, use the Index or Find tabs in the Help system. The SQL Reserved Words / Index topic provides a handy, alphabetized list of key words that may be used for browsing.

Some examples throughout this book are taken from the Microsoft sample database, Nwind.mdb, which is provided on the Manifold CD. This database is for all practical purposes identical to the Northwind.mdb sample database distributed with Microsoft Access and other Microsoft products, except that the Nwind.mdb database uses spaces in the names of some columns ("Order date") where Northwind.mdb does not ("OrderDate").

If we have Access installed on our machine, we have a copy of Northwind.mdb in the C:\Program Files\Microsoft Office\Office10\Samples folder. Use either Nwind.mdb or Northwind.mdb to follow along in the examples. To avoid confusion, both versions of this database are referred to in these examples as the "Northwind" sample database.

If you are working with a downloaded version of Manifold System, consult the manifold.net web site for URLs to download Nwind.mdb and other examples.

Manifold SQL and ANSI Syntax Option

Manifold's SQL is designed to be compatible with Microsoft SQL as implemented in the Microsoft "Jet" database engine. Manifold SQL is intended to be a superset of Jet SQL with features supporting ANSI SQL as well. SQL syntax as documented within the Microsoft Jet SQL Reference for Access 2000 is a practical approach to Manifold SQL.

Manifold SQL has also been extended with special functions to use the spatial and geocoding capabilities of Manifold. Note that the geocoding extensions to SQL will be available only if either the Manifold US street address geocoding database is installed or if Microsoft MapPoint has been installed. See the Spatial Extensions and Geocoding Extensions topics.

In addition to being Microsoft-like, Manifold's SQL engine may be switched into an alternate mode where it parses SQL queries using ANSI syntax. To do so, open the query and in the View - Properties dialog check the Use ANSI-compatible syntax box.
When the ANSI option is turned off (the default), line comments start after an apostrophe (‘) or double dashes (--) and the names of tables and columns can be enclosed in square brackets ([ ]), strings are enclosed in quotes (“”) and times are enclosed in hashes (#).

When the ANSI option is turned on, line comments start after double dashes (--) and the names of tables and columns can be enclosed in square brackets ([ ]), quotes (“” or backward apostrophes (`), strings are enclosed in apostrophes (’) and times are enclosed in hashes (#).

Note from the above that date literals, that is times, are always enclosed in hashes (#). This avoids any possible confusion with string literals regardless of whether the ANSI compatible option is off or on.

Clearly, the safest course given that users might change the settings is to use double dashes (--) to mark line comments and to enclose the names of tables and columns in square brackets ([ ]). The main difference between having the ANSI option off or on is that with it off strings are enclosed in quotes (“”) and with the ANSI option on strings are enclosed in apostrophes (’).

Notes

The idea of relational databases originated in a 1969 IBM research report written by Dr. E.F. "Ted" Codd, who invented the relational model of databases and introduced it to the world. SQL originated as SEQUEL (for "Structured English Query Language") in 1974 at the IBM San Jose Research Laboratory in a project led by Donald Chamberlin.

Just about everyone in computing thinks that "SQL" is an acronym for "Structured Query Language." That’s not true, despite the historical origin of the language, as the ANSI standard defining SQL officially names it "Database Language SQL." Experts will also point out that SQL is not structured, it is used for more purposes than just asking about things (the conventional meaning of the English word "query") and it is technically not a "language" in the Turing sense of the word.

According to ANSI, “SQL” should be pronounced by saying the letters as in “ess cue ell.” However, many people prefer to pronounce the term as the English word “sequel,” especially when referring to Microsoft’s SQL Server DBMS product since “sequel server” is more alliterative and easier to say than “ess cue ell server.” It is very common to use both pronunciations, even in the same sentence, as in “This ess-cue-ell works in Sequel Server but not in Oracle.”

When running SQL within Manifold queries, one is using the Manifold SQL engine. When executing SQL within the Database Console one is using whatever SQL is the native SQL of the external database system. One should be aware that SQL implementations in various database systems can contain bugs. For example, even a very well crafted SQL like Microsoft's Jet SQL contains bugs. If an SQL bug occurs within the Database Console, the bug should be tracked down with the vendor of the external database system being used.

See Also

Working with Queries
Queries
Data Types
Comparison with Jet SQL
Comparison with ANSI SQL
Expressions
Spatial Extensions
Geocoding Extensions
Raster Extensions
The following list includes words reserved by Manifold System for use in SQL statements. See also the names of functions, which are reserved words, in the extensions topics and the Expressions topic.

Future editions of Manifold System could add to the list of the reserved words. Because of this, it is highly recommended that the names of tables or columns used in a query are enclosed in square brackets, so that the query continues to work even if a name of some table or column it refers to becomes a reserved word.

<table>
<thead>
<tr>
<th>Reserved Word</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABS</td>
<td>Expressions (a numeric function)</td>
</tr>
<tr>
<td>ACOS</td>
<td>Expressions (a numeric function)</td>
</tr>
<tr>
<td>ADD</td>
<td>ALTER TABLE Statement</td>
</tr>
<tr>
<td>ADDCOORD</td>
<td>Spatial Extensions (a function)</td>
</tr>
<tr>
<td>ADJACENT</td>
<td>Spatial Extensions (a function)</td>
</tr>
<tr>
<td>ALL</td>
<td>ALL, DISTINCT, SKIP, TOP Quantifiers and ALL, ANY, SOME Quantifiers</td>
</tr>
<tr>
<td>ALLBRANCHES</td>
<td>Aggregate Functions</td>
</tr>
<tr>
<td>ALLCOORDS</td>
<td>Aggregate Functions</td>
</tr>
<tr>
<td>ALTER</td>
<td>ALTER TABLE Statement</td>
</tr>
<tr>
<td>AND</td>
<td>Expressions (a Boolean function) and BETWEEN AND Operator</td>
</tr>
<tr>
<td>ANY</td>
<td>ALL, ANY, SOME Quantifiers</td>
</tr>
<tr>
<td>AREA</td>
<td>Spatial Extensions (a function)</td>
</tr>
<tr>
<td>ARGB</td>
<td>Raster Extensions (a function)</td>
</tr>
<tr>
<td>AS</td>
<td>SELECT Statement and FROM Clause</td>
</tr>
<tr>
<td>ASC</td>
<td>ORDER BY Clause</td>
</tr>
<tr>
<td>ASIN</td>
<td>Expressions (a numeric function)</td>
</tr>
<tr>
<td>ASPECTHEIGHT</td>
<td>Raster Extensions (a function)</td>
</tr>
<tr>
<td>ASSIGNCOORDSYS</td>
<td>Spatial Extensions (a function)</td>
</tr>
<tr>
<td>ATN</td>
<td>Expressions (a numeric function)</td>
</tr>
<tr>
<td>ATN2</td>
<td>Expressions (a numeric function)</td>
</tr>
<tr>
<td>AVG</td>
<td>Aggregate Functions</td>
</tr>
<tr>
<td>BETWEEN</td>
<td>BETWEEN AND Operator</td>
</tr>
<tr>
<td>BGR</td>
<td>Raster Extensions (a function)</td>
</tr>
<tr>
<td>BGRA</td>
<td>Raster Extensions (a function)</td>
</tr>
<tr>
<td>BINARY</td>
<td>Data Types</td>
</tr>
<tr>
<td>BINARYTOHEX</td>
<td>Expressions (a conversion function)</td>
</tr>
<tr>
<td>BIT</td>
<td>Data Types</td>
</tr>
<tr>
<td>BIT_LENGTH</td>
<td>Expressions (a string function)</td>
</tr>
</tbody>
</table>
BLUR        Raster Extensions (a function)
BLURHEIGHT  Raster Extensions (a function)
BORDERBUFFER Spatial Extensions (a function)
BOTH        TRIM Operator
BOUNDARY    Spatial Extensions (a function)
BOUNDINGBOX Spatial Extensions (a function)
BRANCH      Spatial Extensions (a function)
BRANCHCOUNT Spatial Extensions (a function)
BRANCHES    Spatial Extensions (a function)
BRANCHOFF   Spatial Extensions (a function)
BUFFER      Spatial Extensions (a function)
BY          GROUP BY Clause and ORDER BY Clause and SPLIT BY Clause
BYTE        Data Types
CASE         CASE Operator
CAST         CAST Operator
CBOOL        Expressions (a conversion function)
CBYTE        Expressions (a conversion function)
CCOORDSYS   Expressions (a conversion function)
CCUR         Expressions (a conversion function)
CDATE        Expressions (a conversion function)
CDBL         Expressions (a conversion function)
CEIL         Expressions (a numeric function)
CENTROID     Spatial Extensions (a function)
CENTROIDBOX  Spatial Extensions (a function)
CENTROIDINNER Spatial Extensions (a function)
CENTROIDWEIGHT Spatial Extensions (a function)
CENTROIDX    Spatial Extensions (a function)
CENTROIDY    Spatial Extensions (a function)
CGEOM        Expressions (a conversion function)
CGEOMSDE     Expressions (a conversion function)
CGEOMSHP     Expressions (a conversion function)
CGEOMWKB     Expressions (a conversion function)
CHANNEL     Raster Extensions (a function)
CHAR         Data Types
CHAR_LENGTH  Expressions (a string function)
<table>
<thead>
<tr>
<th>Character</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHARACTER</td>
<td>Data Types</td>
</tr>
<tr>
<td>CHARACTER_LENGTH</td>
<td>Expressions (a string function)</td>
</tr>
<tr>
<td>CHR</td>
<td>Expressions (a string function)</td>
</tr>
<tr>
<td>CINT</td>
<td>Expressions (a conversion function)</td>
</tr>
<tr>
<td>CLIPINTERSECT</td>
<td>Spatial Extensions (a function)</td>
</tr>
<tr>
<td>CLIPSUBTRACT</td>
<td>Spatial Extensions (a function)</td>
</tr>
<tr>
<td>CLNG</td>
<td>Expressions (a conversion function)</td>
</tr>
<tr>
<td>CLOSETOADDRESS</td>
<td>Geocoding Extensions (a function)</td>
</tr>
<tr>
<td>CLOSETOZIP</td>
<td>Geocoding Extensions (a function)</td>
</tr>
<tr>
<td>COALESCE</td>
<td>Expressions (a generic function)</td>
</tr>
<tr>
<td>COLUMN</td>
<td>ALTER TABLE Statement</td>
</tr>
<tr>
<td>COMPONENT</td>
<td>Spatial Extensions (used in height functions)</td>
</tr>
<tr>
<td>CONTAINS</td>
<td>Spatial Extensions (a function)</td>
</tr>
<tr>
<td>CONVERTTOAREA</td>
<td>Spatial Extensions (a function)</td>
</tr>
<tr>
<td>CONVERTTOLINE</td>
<td>Spatial Extensions (a function)</td>
</tr>
<tr>
<td>CONVERTTOPOINT</td>
<td>Spatial Extensions (a function)</td>
</tr>
<tr>
<td>CONVEXHULL</td>
<td>Spatial Extensions (a function)</td>
</tr>
<tr>
<td>CONVEXPARTS</td>
<td>Spatial Extensions (a function)</td>
</tr>
<tr>
<td>COORD</td>
<td>Spatial Extensions (a function)</td>
</tr>
<tr>
<td>COORDCOUNT</td>
<td>Spatial Extensions (a function)</td>
</tr>
<tr>
<td>COORDS</td>
<td>Spatial Extensions (a function)</td>
</tr>
<tr>
<td>COORDSYS</td>
<td>Data Types and Spatial Extensions (a function)</td>
</tr>
<tr>
<td>COORDSYSSTOWKT</td>
<td>Spatial Extensions (a function)</td>
</tr>
<tr>
<td>COS</td>
<td>Expressions (a numeric function)</td>
</tr>
<tr>
<td>COUNT</td>
<td>Aggregate Functions</td>
</tr>
<tr>
<td>CREATE</td>
<td>CREATE DRAWING Statement, CREATE TABLE</td>
</tr>
<tr>
<td></td>
<td>Statement and CREATE VIEW Statement</td>
</tr>
<tr>
<td>CROSS</td>
<td>CROSS JOIN Operator</td>
</tr>
<tr>
<td>CSNG</td>
<td>Expressions (a conversion function)</td>
</tr>
<tr>
<td>CSTR</td>
<td>Expressions (a conversion function)</td>
</tr>
<tr>
<td>CURRENCY</td>
<td>Data Types</td>
</tr>
<tr>
<td>CURRENT_DATE</td>
<td>Expressions (a date function)</td>
</tr>
<tr>
<td>CURRENT_TIME</td>
<td>Expressions (a date function)</td>
</tr>
<tr>
<td>CURRENT_TIMESTAMP</td>
<td>Expressions (a date function)</td>
</tr>
<tr>
<td>DATE</td>
<td>Data Types and Expressions (a date function)</td>
</tr>
<tr>
<td>DATEADD</td>
<td>Expressions (a date function)</td>
</tr>
</tbody>
</table>
DATEDIFF Expressions (a date function)
DATEPART Expressions (a date function)
DATESERIAL Expressions (a date function)
DATETIME Data Types
DATEVALUE Expressions (a date function)
DAY Expressions (a date function)
DEFAULT ALTER TABLE Statement, CREATE TABLE Statement and INSERT INTO Statement
DEG2RAD Expressions (a numeric function)
DELETE DELETE Statement
DESC ORDER BY Clause
DIFFERENCEE Raster Extensions (a function)
DIFFERENCEEHEIGHT Raster Extensions (a function)
DIFFERENCEEN Raster Extensions (a function)
DIFFERENCENHEIGHT Raster Extensions (a function)
DIFFERENCENE Raster Extensions (a function)
DIFFERENCENEHEIGHT Raster Extensions (a function)
DIFFERENCENW Raster Extensions (a function)
DIFFERENCENWHIGH Raster Extensions (a function)
DIFFERENCES Raster Extensions (a function)
DIFFERENCESHEIGHT Raster Extensions (a function)
DIFFERENCESE Raster Extensions (a function)
DIFFERENCESEHEIGHT Raster Extensions (a function)
DIFFERENCESW Raster Extensions (a function)
DIFFERENCESWHEIGHT Raster Extensions (a function)
DIFFERENCEW Raster Extensions (a function)
DIFFERENCEWHEIGHT Raster Extensions (a function)
DISTANCE Spatial Extensions (a function)
DISTANCEEARTH Spatial Extensions (a function)
DISTANCETOADRESS Geocoding Extensions (a function)
DISTANCETOZIP Geocoding Extensions (a function)
DISTINCT ALL, DISTINCT, SKIP, TOP Quantifiers
DISTINCTROW ALL, DISTINCT, SKIP, TOP Quantifiers
DISTINCTROW is identical to DISTINCT.
DIV Expressions (a numeric operator)
DOUBLE Data Types
SQL Reference Guide

DRAWING  CREATE DRAWING Statement
DROP  DROP TABLE Statement, DROP VIEW Statement and ALTER TABLE Statement
ELSE  CASE Operator
ENCLOSINGCIRCLE  Spatial Extensions (a function)
ENCLOSINGRECTANGLE  Spatial Extensions (a function)
END  CASE Operator
ENDPOINT  Spatial Extensions (a function)
EQV  Expressions (a Boolean operator)
EXCEPT  EXCEPT Operator
EXISTS  EXISTS Operator
EXP  Expressions (a numeric function)
FALSE  IS Operator and Expressions (a Boolean operator)
FIRST  Aggregate Functions
FIX  Expressions (a numeric function)
FLIPHORIZONTALLY  Spatial Extensions (a function)
FLIPVERTICALLY  Spatial Extensions (a function)
FOR  SUBSTRING Operator
FORMATCURRENCY  Expressions (a string function)
FORMATDATETIME  Expressions (a string function)
FORMATNUMBER  Expressions (a string function)
FORMATPERCENT  Expressions (a string function)
FROM  FROM Clause
FULL  OUTER (LEFT, RIGHT, FULL) JOIN Operators
GEOM  Data Types
GEOMSDE  Data Types
GEOMSHP  Data Types
GEOMTYPE  Spatial Extensions (a function)
GEOMWKB  Data Types
GRAVITY  Raster Extensions (a function)
GROUP  GROUP BY Clause
HASPIXEL  Raster Extensions (a function)
HAVING  HAVING Clause
HEIGHT  Raster Extensions (a function)
HEIGHTAVG  Raster Extensions (a function)
HEIGHTCURVMEAN  Raster Extensions (a function)
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HEIGHTCURVPLAN</td>
<td>Raster Extensions (a function)</td>
</tr>
<tr>
<td>HEIGHTCURVPROFILE</td>
<td>Raster Extensions (a function)</td>
</tr>
<tr>
<td>HEIGHTDIVERSITY</td>
<td>Raster Extensions (a function)</td>
</tr>
<tr>
<td>HEIGHTDIVERSITYINDEX</td>
<td>Raster Extensions (a function)</td>
</tr>
<tr>
<td>HEIGHTMAJ</td>
<td>Raster Extensions (a function)</td>
</tr>
<tr>
<td>HEIGHTMAX</td>
<td>Raster Extensions (a function)</td>
</tr>
<tr>
<td>HEIGHTMED</td>
<td>Raster Extensions (a function)</td>
</tr>
<tr>
<td>HEIGHTMIN</td>
<td>Raster Extensions (a function)</td>
</tr>
<tr>
<td>HEIGHTSUM</td>
<td>Raster Extensions (a function)</td>
</tr>
<tr>
<td>HEX</td>
<td>Expressions (a string function)</td>
</tr>
<tr>
<td>HEXTOBINARY</td>
<td>Expressions (a conversion function)</td>
</tr>
<tr>
<td>HIGHPASS1</td>
<td>Raster Extensions (a function)</td>
</tr>
<tr>
<td>HIGHPASS1HEIGHT</td>
<td>Raster Extensions (a function)</td>
</tr>
<tr>
<td>HIGHPASS2</td>
<td>Raster Extensions (a function)</td>
</tr>
<tr>
<td>HIGHPASS2HEIGHT</td>
<td>Raster Extensions (a function)</td>
</tr>
<tr>
<td>HIGHPASS3</td>
<td>Raster Extensions (a function)</td>
</tr>
<tr>
<td>HIGHPASS3HEIGHT</td>
<td>Raster Extensions (a function)</td>
</tr>
<tr>
<td>HLS</td>
<td>Raster Extensions (a function)</td>
</tr>
<tr>
<td>HOUR</td>
<td>Expressions (a date function)</td>
</tr>
<tr>
<td>HUE</td>
<td>Raster Extensions (a function)</td>
</tr>
<tr>
<td>IIF</td>
<td>Expressions (a generic function)</td>
</tr>
<tr>
<td>IMP</td>
<td>Expressions (a Boolean operator)</td>
</tr>
<tr>
<td>IN</td>
<td>IN Operator, POSITION Operator and TRANSFORM Statement</td>
</tr>
<tr>
<td>INNER</td>
<td>INNER JOIN Operator</td>
</tr>
<tr>
<td>INNERBUFFER</td>
<td>Spatial Extensions (a function)</td>
</tr>
<tr>
<td>INSERT</td>
<td>INSERT INTO Statement</td>
</tr>
<tr>
<td>INSTR</td>
<td>Expressions (a string function)</td>
</tr>
<tr>
<td>INSTRREV</td>
<td>Expressions (a string function)</td>
</tr>
<tr>
<td>INT</td>
<td>Data Types and Expressions (a numeric function)</td>
</tr>
<tr>
<td>INTEGER</td>
<td>Data Types</td>
</tr>
<tr>
<td>INTENSITY</td>
<td>Raster Extensions (a function)</td>
</tr>
<tr>
<td>INTERPOLATEHEIGHT</td>
<td>Raster Extensions (a function)</td>
</tr>
<tr>
<td>INTERPOLATEROWHEIGHT</td>
<td>Raster Extensions (a function)</td>
</tr>
<tr>
<td>INTERSECT</td>
<td>INTERSECT Operator</td>
</tr>
<tr>
<td>INTERSECTIONPOINT</td>
<td>Spatial Extensions (a function)</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>INTERSECTLINE</td>
<td>Spatial Extensions (a function)</td>
</tr>
<tr>
<td>INTERSECTS</td>
<td>Spatial Extensions (a function)</td>
</tr>
<tr>
<td>INTO</td>
<td>INSERT INTO Statement, SELECT INTO Statement</td>
</tr>
<tr>
<td>IS</td>
<td>IS Operator and IS NULL Operator</td>
</tr>
<tr>
<td>ISAREA</td>
<td>Spatial Extensions (a function)</td>
</tr>
<tr>
<td>ISCLOSED</td>
<td>Spatial Extensions (a function)</td>
</tr>
<tr>
<td>ISLANDS</td>
<td>Spatial Extensions (a function)</td>
</tr>
<tr>
<td>ISLINE</td>
<td>Spatial Extensions (a function)</td>
</tr>
<tr>
<td>ISPOINT</td>
<td>Spatial Extensions (a function)</td>
</tr>
<tr>
<td>ISRING</td>
<td>Spatial Extensions (a function)</td>
</tr>
<tr>
<td>JOIN</td>
<td>CROSS JOIN Operator, INNER JOIN Operator and OUTER (LEFT, RIGHT, FULL) JOIN Operators</td>
</tr>
<tr>
<td>JOINLINES</td>
<td>Spatial Extensions (a function)</td>
</tr>
<tr>
<td>KRINGING</td>
<td>Raster Extensions (a function)</td>
</tr>
<tr>
<td>LAPLACE1</td>
<td>Raster Extensions (a function)</td>
</tr>
<tr>
<td>LAPLACE1HEIGHT</td>
<td>Raster Extensions (a function)</td>
</tr>
<tr>
<td>LAPLACE2</td>
<td>Raster Extensions (a function)</td>
</tr>
<tr>
<td>LAPLACE2HEIGHT</td>
<td>Raster Extensions (a function)</td>
</tr>
<tr>
<td>LAST</td>
<td>Aggregate Functions</td>
</tr>
<tr>
<td>LATITUDE</td>
<td>Data Types</td>
</tr>
<tr>
<td>LCASE</td>
<td>Expressions (a string function)</td>
</tr>
<tr>
<td>LEADING</td>
<td>TRIM Operator</td>
</tr>
<tr>
<td>LEAVING</td>
<td>LEAVING Clause</td>
</tr>
<tr>
<td>LEFT</td>
<td>OUTER (LEFT, RIGHT, FULL) JOIN Operators and Expressions (a string function)</td>
</tr>
<tr>
<td>LEN</td>
<td>Expressions (a string function)</td>
</tr>
<tr>
<td>LENGTH</td>
<td>Spatial Extensions (a function)</td>
</tr>
<tr>
<td>LIGHTNESS</td>
<td>Raster Extensions (a function)</td>
</tr>
<tr>
<td>LIKE</td>
<td>LIKE Operator</td>
</tr>
<tr>
<td>LIKEX</td>
<td>LIKEX Operator</td>
</tr>
<tr>
<td>LINEPART</td>
<td>Spatial Extensions (a function)</td>
</tr>
<tr>
<td>LINEPOINT</td>
<td>Spatial Extensions (a function)</td>
</tr>
<tr>
<td>LOCATEADDRESS</td>
<td>Geocoding Extensions (a function)</td>
</tr>
<tr>
<td>LOCATEADDRESSLAT</td>
<td>Geocoding Extensions (a function)</td>
</tr>
<tr>
<td>LOCATEADDRESSLON</td>
<td>Geocoding Extensions (a function)</td>
</tr>
<tr>
<td>LOCATEZIP</td>
<td>Geocoding Extensions (a function)</td>
</tr>
<tr>
<td>LOCATEZIPLAT</td>
<td>Geocoding Extensions (a function)</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>LOCATEZIPLO</td>
<td>Geocoding Extensions (a function)</td>
</tr>
<tr>
<td>LOG</td>
<td>Expressions (a numeric function)</td>
</tr>
<tr>
<td>LOG10</td>
<td>Expressions (a numeric function)</td>
</tr>
<tr>
<td>LOG2</td>
<td>Expressions (a numeric function)</td>
</tr>
<tr>
<td>LONG</td>
<td>Data Types</td>
</tr>
<tr>
<td>LONGBINARY</td>
<td>Data Types</td>
</tr>
<tr>
<td>LONGITUDE</td>
<td>Data Types</td>
</tr>
<tr>
<td>LONGTEXT</td>
<td>Data Types</td>
</tr>
<tr>
<td>LOWER</td>
<td>Expressions (a string function)</td>
</tr>
<tr>
<td>LOWPASS1</td>
<td>Raster Extensions (a function)</td>
</tr>
<tr>
<td>LOWPASS1HEIGHT</td>
<td>Raster Extensions (a function)</td>
</tr>
<tr>
<td>LOWPASS2</td>
<td>Raster Extensions (a function)</td>
</tr>
<tr>
<td>LOWPASS2HEIGHT</td>
<td>Raster Extensions (a function)</td>
</tr>
<tr>
<td>LOWPASS3</td>
<td>Raster Extensions (a function)</td>
</tr>
<tr>
<td>LOWPASS3HEIGHT</td>
<td>Raster Extensions (a function)</td>
</tr>
<tr>
<td>LTRIM</td>
<td>Expressions (a string function)</td>
</tr>
<tr>
<td>MAX</td>
<td>Aggregate Functions</td>
</tr>
<tr>
<td>MAXX</td>
<td>Spatial Extensions (a function)</td>
</tr>
<tr>
<td>MAXY</td>
<td>Spatial Extensions (a function)</td>
</tr>
<tr>
<td>MEDIANCROSS</td>
<td>Raster Extensions (a function)</td>
</tr>
<tr>
<td>MEDIANCROSSHEIGHT</td>
<td>Raster Extensions (a function)</td>
</tr>
<tr>
<td>MEDIANSQUARE</td>
<td>Raster Extensions (a function)</td>
</tr>
<tr>
<td>MEDIANSQUAREHEIGHT</td>
<td>Raster Extensions (a function)</td>
</tr>
<tr>
<td>MEDIANSQUARE5</td>
<td>Raster Extensions (a function)</td>
</tr>
<tr>
<td>MEDIANSQUARESHEIGHT</td>
<td>Raster Extensions (a function)</td>
</tr>
<tr>
<td>MID</td>
<td>Expressions (a string function)</td>
</tr>
<tr>
<td>MIN</td>
<td>Aggregate Functions</td>
</tr>
<tr>
<td>MINUTE</td>
<td>Expressions (a date function)</td>
</tr>
<tr>
<td>MINX</td>
<td>Spatial Extensions (a function)</td>
</tr>
<tr>
<td>MINY</td>
<td>Spatial Extensions (a function)</td>
</tr>
<tr>
<td>MOD</td>
<td>Expressions (a numeric operator)</td>
</tr>
<tr>
<td>MONEY</td>
<td>Data Types</td>
</tr>
<tr>
<td>MONTH</td>
<td>Expressions (a date function)</td>
</tr>
<tr>
<td>MONTHNAME</td>
<td>Expressions (a date function)</td>
</tr>
<tr>
<td>MOVEHORIZONTALLY</td>
<td>Spatial Extensions (a function)</td>
</tr>
<tr>
<td>MOVEVERTICALLY</td>
<td>Spatial Extensions (a function)</td>
</tr>
</tbody>
</table>
NAMES

NEWLINE Spatial Extensions (a function)

NEWPOINT Spatial Extensions (a function)

NEWPOINTLATLON Spatial Extensions (a function)

NO Expressions (a Boolean literal)

NORMALIZE Spatial Extensions (a function)

NOT IS Operator , IS NULL Operator , BETWEEN AND Operator , IN Operator , LIKE Operator , LIKEX Operator and Expressions (a Boolean operator)

NOW Expressions (a date function)

NULL ALTER TABLE Statement , CREATE TABLE Statement and IS NULL Operator

NULLIF Expressions (a generic function)

OCT Expressions (a string function)

OCTET_LENGTH Expressions (a string function)

ON INNER JOIN Operator and OUTER (LEFT, RIGHT, FULL) JOIN Operators

OPTIONS OPTIONS Clause

OR Expressions (a Boolean operator)

ORDER ORDER BY Clause

OUTER OUTER (LEFT, RIGHT, FULL) JOIN Operators

PARAMETERS PARAMETERS Declaration

PERCENT ALL, DISTINCT, SKIP, TOP Quantifiers

PIVOT TRANSFORM Statement

PIXEL Raster Extensions (a function)

PIXELHEIGHT Raster Extensions (a function)

PIXELSBYX Raster Extensions (a function)

PIXELSBYY Raster Extensions (a function)

PIXELWIDTH Raster Extensions (a function)

POSITION POSITION Operator

POW Expressions (a numeric function)

PRESET Spatial Extensions (used in coordinate system functions)

PROJECT Spatial Extensions (a function)

RAD2DEG Expressions (a numeric function)

REAL Data Types

RECTHEIGHT Spatial Extensions (a function)

RECTWIDTH Spatial Extensions (a function)

REEXP Expressions (a string function)
<table>
<thead>
<tr>
<th>Function</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>RENAME</td>
<td>ALTER TABLE Statement</td>
</tr>
<tr>
<td>REPLACE</td>
<td>Expressions (a string function)</td>
</tr>
<tr>
<td>REVERSELINE</td>
<td>Spatial Extensions (a function)</td>
</tr>
<tr>
<td>RGB</td>
<td>Expressions (a numeric function)</td>
</tr>
<tr>
<td>RIGHT</td>
<td>Expressions (a string function)</td>
</tr>
<tr>
<td>RND</td>
<td>Expressions (a numeric function)</td>
</tr>
<tr>
<td>ROTATE</td>
<td>Spatial Extensions (a function)</td>
</tr>
<tr>
<td>ROUND</td>
<td>Expressions (a numeric function)</td>
</tr>
<tr>
<td>RTRIM</td>
<td>Expressions (a string function)</td>
</tr>
<tr>
<td>SATURATION</td>
<td>Raster Extensions (a function)</td>
</tr>
<tr>
<td>SCALE</td>
<td>Spatial Extensions (a function)</td>
</tr>
<tr>
<td>SCALEHORizontally</td>
<td>Spatial Extensions (a function)</td>
</tr>
<tr>
<td>SCALEVertically</td>
<td>Spatial Extensions (a function)</td>
</tr>
<tr>
<td>SECOND</td>
<td>Expressions (a date function)</td>
</tr>
<tr>
<td>SEGMENTS</td>
<td>Spatial Extensions (a function)</td>
</tr>
<tr>
<td>SELECT</td>
<td>SELECT Statement and SELECT … INTO Statement</td>
</tr>
<tr>
<td>SET</td>
<td>ALTER TABLE Statement and UPDATE Statement</td>
</tr>
<tr>
<td>SGN</td>
<td>Expressions (a numeric function)</td>
</tr>
<tr>
<td>SHAPEHULL</td>
<td>Spatial Extensions (a function)</td>
</tr>
<tr>
<td>SHARPEN</td>
<td>Raster Extensions (a function)</td>
</tr>
<tr>
<td>SHARPENHEIGHT</td>
<td>Raster Extensions (a function)</td>
</tr>
<tr>
<td>SHARPENMORE</td>
<td>Raster Extensions (a function)</td>
</tr>
<tr>
<td>SHARPENMOREHEIGHT</td>
<td>Raster Extensions (a function)</td>
</tr>
<tr>
<td>SHORT</td>
<td>Data Types</td>
</tr>
<tr>
<td>SIN</td>
<td>Expressions (a numeric function)</td>
</tr>
<tr>
<td>SINGLE</td>
<td>Data Types</td>
</tr>
<tr>
<td>SKIP</td>
<td>ALL, DISTINCT, SKIP, TOP Quantifiers</td>
</tr>
<tr>
<td>SLOPEHEIGHT</td>
<td>Raster Extensions (a function)</td>
</tr>
<tr>
<td>SMALLINT</td>
<td>Data Types</td>
</tr>
<tr>
<td>SOME</td>
<td>ALL, ANY, SOME Quantifiers</td>
</tr>
<tr>
<td>SOUNDSLIKE</td>
<td>Expressions (a string function)</td>
</tr>
<tr>
<td>SPACE</td>
<td>Expressions (a string function)</td>
</tr>
<tr>
<td>SPLIT BY</td>
<td>SPLIT BY Clause</td>
</tr>
<tr>
<td>SQR</td>
<td>Expressions (a numeric function)</td>
</tr>
<tr>
<td>STARTPOINT</td>
<td>Spatial Extensions (a function)</td>
</tr>
</tbody>
</table>
STDEV Aggregate Functions
STDEVP Aggregate Functions
STRCOMP Expressions (a string function)
STRING Expressions (a string function)
STREVERSE Expressions (a string function)
SUBSTRING SUBSTRING Operator
SUM Aggregate Functions
TABLE ALTER TABLE Statement, CREATE TABLE Statement, DROP TABLE Statement and SELECT Statement
TAN Expressions (a numeric function)
TEXT Data Types
THEN CASE Operator
TILE Raster Extensions (a function)
TILEHEIGHT Raster Extensions (a function)
TILEMEDIAN Raster Extensions (a function)
TILEMEDIANHEIGHT Raster Extensions (a function)
TIME Data Types and Expressions (a date function)
TIMER Expressions (a date function)
TIMESERIAL Expressions (a date function)
TIMESTAMP Data Types
TIMEVALUE Expressions (a date function)
TINYINT Data Types
TO ALTER TABLE Statement
TOKEN Expressions (a string function)
TOP ALL, DISTINCT, SKIP, TOP Quantifiers
TOUCHES Spatial Extensions (a function)
TRAILING TRIM Operator
TRANSFORM TRANSFORM Statement
TRIANGLES Spatial Extensions (a function)
TRIANGULATION Spatial Extensions (a function)
TRIANGULATIONLINES Spatial Extensions (a function)
TRIM TRIM Operator
TRUE IS Operator and Expressions (a Boolean literal)
TYPENAME Expressions (a generic function)
UCASE Expressions (a string function)
UNION UNION Operator
UNIONALL  Spatial Extensions (a function)
UNIONAREAS Spatial Extensions (a function)
UNIQUE   UNIQUE Operator
UNKNOWN  IS Operator
UPDATE   UPDATE Statement
UPPER    Expressions (a string function)
VALUES   SELECT Statement and INSERT INTO Statement
Var      Aggregate Functions
VARCHAR  Data Types
VARP     Aggregate Functions
VARTYPE  Expressions (a generic function)
VIEW     CREATE VIEW Statement and DROP VIEW Statement
VOLUME   Raster Extensions (a function)
VORONOI  Spatial Extensions (a function)
VORONOILINES Spatial Extensions (a function)
VORONOIPOINTS Spatial Extensions (a function)
WEEKDAY  Expressions (a date function)
WEEKDAYNAME Expressions (a date function)
WHEN     CASE Operator
WHERE    WHERE Clause
WKTTOCOORDSYS Spatial Extensions (a function)
XOR      Expressions (a Boolean operator)
YEAR     Expressions (a date function)
YES      Expressions (a Boolean literal)

See Also

Spatial Extensions
Geocoding Extensions
Raster Extensions
Expressions
Data Types

Manifold SQL data types consist of several primary data types and several valid synonyms recognized for these data types.

The following table lists the primary data types. The synonyms are identified in the SQL Reserved Words / Index topic.

<table>
<thead>
<tr>
<th>Data type</th>
<th>Storage size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BINARY</td>
<td>1 byte per character</td>
<td>Any type of data may be stored in a column of this type. No translation of the data (for example, to text) is made. How the data is input in a binary column dictates how it will appear as output.</td>
</tr>
<tr>
<td>BIT</td>
<td>1 byte</td>
<td>Yes and No values and columns that contain only one of two values.</td>
</tr>
<tr>
<td>BYTE</td>
<td>1 byte</td>
<td>An integer value between 0 and 255.</td>
</tr>
<tr>
<td>COORDSYS</td>
<td>variable</td>
<td>A coordinate system token.</td>
</tr>
<tr>
<td>CURRENCY</td>
<td>8 bytes</td>
<td>A scaled integer between -9.2E14 and 9.2E14.</td>
</tr>
<tr>
<td>DATETIME</td>
<td>16 bytes</td>
<td>A date or time value between the years 100 and 9999.</td>
</tr>
<tr>
<td>DOUBLE</td>
<td>8 bytes</td>
<td>A double-precision floating-point value with a range of -1.8E308 to 1.8E308</td>
</tr>
<tr>
<td>GEOM</td>
<td>variable</td>
<td>A geometric object, includes coordinate system data.</td>
</tr>
<tr>
<td>GEOMSDE</td>
<td>variable</td>
<td>A geometric object in ESRI ArcSDE format.</td>
</tr>
<tr>
<td>GEOMSHP</td>
<td>variable</td>
<td>A geometric object in ESRI SHP format (geodatabase).</td>
</tr>
<tr>
<td>GEOMWKB</td>
<td>variable</td>
<td>A geometric object in OGC WKB format.</td>
</tr>
<tr>
<td>LATITUDE</td>
<td>8 bytes</td>
<td>A double-precision floating-point latitude value.</td>
</tr>
<tr>
<td>LONG</td>
<td>4 bytes</td>
<td>A long integer between -2,147,483,648 and 2,147,483,647.</td>
</tr>
<tr>
<td>LONGITUDE</td>
<td>8 bytes</td>
<td>A double-precision floating-point longitude value.</td>
</tr>
<tr>
<td>SHORT</td>
<td>2 bytes</td>
<td>A short integer between -32,768 and 32,767.</td>
</tr>
<tr>
<td>SINGLE</td>
<td>4 bytes</td>
<td>A single-precision floating-point value with a range of -3.4E38 to 3.4E38.</td>
</tr>
<tr>
<td>TEXT</td>
<td>2 bytes per character (Unicode)</td>
<td>Zero to a maximum of 2 gigabytes.</td>
</tr>
</tbody>
</table>

Synonyms

The following table lists synonym names for the primary data types used in Manifold SQL:

<table>
<thead>
<tr>
<th>Synonym</th>
<th>Data Type</th>
</tr>
</thead>
</table>

String Literals

SQL is case insensitive for SQL itself but is case sensitive when evaluating string values. Suppose, for example, we have a query such as:

```
SELECT [F] FROM [Table] WHERE [F] = "13AF";
```

This query will return no results if the value in "F" is "13aF", because that is not the same as "13AF".

To normalize case, use the Transform toolbar Make Lower Case or Make Upper Case operators to force all values in the column to upper case or lower case. Alternatively, use a modified SELECT statement:

```
SELECT [F] FROM [Table] WHERE UCase([F]) = "13AF";
```

When comparing strings, SQL is sensitive to leading or trailing whitespace. "13AF " does not equal "13AF". To remove leading / trailing whitespace use the Transform toolbar Trim Left, Trim Right or Trim operators to eliminate leading, trailing or both leading and trailing whitespace. Alternatively, use a modified SELECT statement:

```
SELECT [F] FROM [Table] WHERE Trim([F]) = "13AF";
```

The strings are enclosed either in apostrophes (') or in quotes (") depending on whether the query is ANSI-compatible or not.

Date Literals

The date literals used in SQL must use either ANSI format (year-month-day hour:minute:second) or one of the formats mentioned in the Regional Options applet in the Windows Control Panel.

To find records dated July 10, 1992 on a machine with US regional options, use either of the following statements:

```
SELECT * FROM [Orders] WHERE [Shipped Date] = #92-7-10#;
SELECT * FROM [Orders] WHERE [Shipped Date] = #7/10/92#;
```
The dates are enclosed in hash signs (#). They can also be enclosed in apostrophes (') or in quotes (") depending on whether the query is ANSI-compatible or not.

See Also

Expressions
Spatial Extensions
Geocoding Extensions
Raster Extensions
Spatial Extensions

Manifold System provides additional functions within SQL for selecting objects in drawings based on their geometric properties. Manifold also includes SQL functions to manipulate geometry data saved within tables.

The following table lists available spatial functions. Functions are often employed with data in tables of type Geometry, Geometry (SDE), Geometry (SHP) or Geometry (WKB). Different geometry types have different limitations. For example, the Geometry (SDE) type does not distinguish between lines and areas and uses integer coordinate values, and none of the geometry types except Geometry stores coordinate system data.

See the Geometry in Tables and the Queries and Geoms topics for additional information on storing and using geometry in tables.

<table>
<thead>
<tr>
<th>Function</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddCoord(line, point)</td>
<td>Adds the given coordinate to the first branch of the given line and returns the result.</td>
</tr>
<tr>
<td>Adjacent(geom, geom)</td>
<td>Returns True if the objects share at least one common location, and all common locations shared by the objects lie on a boundary of both objects; otherwise returns False.</td>
</tr>
<tr>
<td>AllBranches(geom)</td>
<td>Works on geoms of the same type and creates a geom with all branches of all source geoms. An aggregate function.</td>
</tr>
<tr>
<td>AllCoords(geom)</td>
<td>Works on geoms of any type and creates a multipoint geom with all coordinates (inflection points) of all source geoms. An aggregate function.</td>
</tr>
<tr>
<td>Area(geom[, unit])</td>
<td>Returns the area of the object in the specified units.</td>
</tr>
<tr>
<td>AreaEarth(geom[, unit])</td>
<td>Returns the ellipsoidal area of the object in the specified metric units.</td>
</tr>
<tr>
<td>AssignCoordSys(geom, coordsys)</td>
<td>Assigns a coordinate system to a geom and returns the result.</td>
</tr>
<tr>
<td>BorderBuffer(geom, width[, unit])</td>
<td>Returns border buffer of specified width for given area.</td>
</tr>
<tr>
<td>Boundary(geom)</td>
<td>Returns boundary of given area.</td>
</tr>
<tr>
<td>BoundingBox(geom)</td>
<td>Returns bounding box of given geometric object.</td>
</tr>
<tr>
<td>Branch(geom, index)</td>
<td>Returns specified branch of given geometric object.</td>
</tr>
<tr>
<td>BranchCount(geom)</td>
<td>Returns number of branches in given geometric object.</td>
</tr>
<tr>
<td>Branches(geom)</td>
<td>Splits geometric object into branches. A split function.</td>
</tr>
<tr>
<td>BranchOf(geom, index)</td>
<td>Returns number of branch in given geometric object that contains point with specified index.</td>
</tr>
<tr>
<td>Buffer(geom, width[, unit])</td>
<td>Returns buffer of specified width for given geometric object.</td>
</tr>
<tr>
<td>CentroidX(geom)</td>
<td>Returns the X value of the object's centroid coordinates.</td>
</tr>
<tr>
<td>CentroidY(geom)</td>
<td>Returns the Y value of the object's centroid coordinates.</td>
</tr>
<tr>
<td>Function Description</td>
<td></td>
</tr>
<tr>
<td>----------------------</td>
<td></td>
</tr>
<tr>
<td>ClipIntersect(geom, clipWith)</td>
<td>Returns parts of given geometric object lying within clipping area.</td>
</tr>
<tr>
<td>ClipSubtract(geom, clipWith)</td>
<td>Returns parts of given geometric object lying outside clipping area.</td>
</tr>
<tr>
<td>Contains(geom, geom)</td>
<td>Returns True if the first object entirely contains the second object.</td>
</tr>
<tr>
<td>ConvexHull(geom)</td>
<td>Converts a line or a point to a point set.</td>
</tr>
<tr>
<td>ConvexParts(geom[, tolerance])</td>
<td>Decomposes given area to convex parts. The tolerance parameter is in the coordinate system units of the geom.</td>
</tr>
<tr>
<td>Coord(geom, index)</td>
<td>Returns coord (inflection point) of given geometric object with specified index.</td>
</tr>
<tr>
<td>CoordCount(geom)</td>
<td>Returns total number of coords (inflection points) in all branches of given geometric object.</td>
</tr>
<tr>
<td>Coords(geom)</td>
<td>Splits geometric object into coordinates. A split function.</td>
</tr>
<tr>
<td>CoordSys(system [AS COMPONENT])</td>
<td>Loads a coordinate system from a component or a preset.</td>
</tr>
<tr>
<td>CoordSysToWKT(coordsys)</td>
<td>Prints a coordinate system into a WKT (PRJ) string.</td>
</tr>
<tr>
<td>Distance(geom, geom[, unit])</td>
<td>Returns the distance between the two objects in the specified units.</td>
</tr>
<tr>
<td>DistanceEarth(geomA, geomB [, unit])</td>
<td>Returns ellipsoidal distance between given points.</td>
</tr>
<tr>
<td>EnclosingCircle(geom)</td>
<td>Returns minimum circle containing given geometric object.</td>
</tr>
<tr>
<td>EnclosingRectangle(geom)</td>
<td>Returns minimum rectangle containing given geometric object.</td>
</tr>
<tr>
<td>EndPoint(geom)</td>
<td>Returns last point in given line that contains one branch.</td>
</tr>
<tr>
<td>FlipHorizontally(geom)</td>
<td>Flips given geometric object horizontally and returns the result.</td>
</tr>
<tr>
<td>FlipVertically(geom)</td>
<td>Flips given geometric object vertically and returns the result.</td>
</tr>
<tr>
<td>GeomType(geom)</td>
<td>Returns type of given geometric object as a number.</td>
</tr>
<tr>
<td>InnerBuffer(geom, width[, unit])</td>
<td>Returns inner buffer of specified width for given area.</td>
</tr>
<tr>
<td>IntersectionPoint(line, lineOther)</td>
<td>Intersects given line with another geometric object by splitting it into branches at intersection points and returns the results.</td>
</tr>
<tr>
<td>Intersects(geom, geom)</td>
<td>For points, returns True if a point belongs to another object, otherwise returns False. For areas and lines, returns True if the objects share at least one common location that does not lie on a boundary of both objects and if neither object contains the other object; otherwise returns False.</td>
</tr>
</tbody>
</table>
IsArea(geom)  Returns True if the object is an area; otherwise returns False.

IsClosed(geom)  Returns True if given line is closed and False otherwise.

Islands(geom)  Splits area into islands. An island is a subarea which consists of an outer contour and zero or more inner contours (holes). A split function.

IsLine(geom)  Returns True if the object is a line; otherwise returns False.

IsPoint(geom)  Returns True if the object is a point; otherwise returns False.

IsRing(geom)  Returns True if given line is closed and contains one (and only one) branch with no self-intersections and False otherwise.

JoinLines(line, lineOther)  Joins two given lines and returns the result.

Length(geom[, unit])  Returns the length or the perimeter of the object in the specified units.

LengthEarth(geom[, unit])  Returns the ellipsoidal length or the perimeter of the object in the specified metric units.

LinePart(geom, f, t[, unit])  Returns that part of a line between specified distance limits on given line. Both distances are measured from the start of the line and continue between branches. f is the "from" distance and t is the "to" distance for the new line. Suppose we have a road that is 12 miles long. LinePart(0, 6, "mi") will return the first half of the road, and LinePart(12, 6, "mi") will return the second half of the road reversed in backward direction. If a unit is not specified, the default native unit of the geom is used.

LinePoint(geom, d[, unit])  Returns a point that lies at the specified distance on given line. The distance is measured from the start of line and continues between branches.

MaxX(geom)  Returns the maximum X value in the object's coordinates.

MaxY(geom)  Returns the maximum Y value in the object's coordinates.

MinX(geom)  Returns the minimum X value in the object's coordinates.

MinY(geom)  Returns the minimum Y value in the object's coordinates.

MoveHorizontally(geom, distance[, unit])  Moves given geometric object by specified distance horizontally and returns the result.

MoveVertically(geom, distance[, unit])  Moves given geometric object by specified distance vertically and returns the result.

NewLine(geom, geom[, geom ...])  Returns a line using two or more point geom arguments or two or more line geom arguments. The coordinate system of the line equals that of the first geom. If all of the arguments are point geoms, the result is a line with a single branch from the first point to the last. If all of the arguments are lines, the result is a line with multiple branches. Otherwise (for example, a mix of point and line geoms), the result is a NULL.

NewPoint(x, y)  Returns a point with given XY coordinates in default coordinate system.

NewPointLatLon(x, y)  Returns a point with given XY coordinates in lat/lon coordinate system.

Normalize(geom)  Normalizes metric of given geometric object and returns the result.
Project(geom, system)  Project given geometric object to another coordinate system.

RectHeight(geom[, unit]) Returns the height of the object bounding box in the specified units.

RectWidth(geom[, unit]) Returns the width of the object bounding box for the object in the specified units.

ReverseLine(geom) Reverses direction of all branches in given line and returns the result.

Rotate(geom, angle) Rotates given geometric object by specified amount of degrees and returns the result.

RotateAbout(geom, center, angle) Rotates given geometric object, geom, about another geometric object, the center, by specified amount in degrees in angle and returns the result. If the object given as the center is a line or area the centroid of the object will be used for the axis of rotation.

Scale(geom, factor) Scales given geometric object by specified factor and returns the result.

ScaleHorizontally(geom, factor) Scales given geometric object by specified factor horizontally and returns the result.

ScaleVertically(geom, factor) Scales given geometric object by specified factor vertically and returns the result.

Segments(geom, subsegments) Splits each segment in given line or area into specified number of subsegments and returns the result.

ShapeHull(geom, cells) Returns shape hull of given geometric object.

StartPoint(geom) Returns first point in given line that contains one branch.

Touches(geom, geom) Returns True if the objects share at least one common location; otherwise returns False.

Triangles(geom[, tolerance]) Returns a triangulation of the given area. The tolerance parameter is in the coordinate system units of the geom.

Triangulation(geom[, tolerance]) Returns areas in the triangulation of a point set. The tolerance parameter is in the coordinate system units of the geom. See the Transform - Triangulation topic for a discussion of triangulation.

TriangulationLines(geom[, tolerance]) Returns lines in the triangulation of a point set. The tolerance parameter is in the coordinate system units of the geom. See the Transform - Triangulation topic for a discussion of triangulation.

UnionAll(geom) Unions all areas. An aggregate function.

UnionAreas(geom, geom) Unions two given areas and returns the result.

Voronoii(geom[, tolerance]) Returns areas in the Voronoi diagram of a point set. The tolerance parameter is in the coordinate system units of the geom. See the Transform - Voronoi Operators topic for a discussion of Voronoi diagrams (also called Dirichlet or Thiessen tessellations by some cultures).

VoronoiiLines(geom[, tolerance]) Returns lines in the Voronoi diagram of a point set. The tolerance parameter is in the coordinate system units of the geom.

VoronoiiPoints(geom[, tolerance]) Returns points in the Voronoi diagram of a point set. The tolerance parameter is in the coordinate system units of the geom.

WKTTToCoordSys(wkt) Parses a WKT (PRJ) string into a coordinate system.
See the Expressions topic for additional SQL expressions.

**Aggregate Functions**

Queries can use the AllBranches, AllCoords and UnionAll aggregate functions to aggregate Geom columns.

Importing or linking a geom created by AllBranches creates one line, one area, or one or more points. Importing or linking a geom created by AllCoords creates one point at the location of the first coordinate. To import the value returned by AllCoords as a set of points, alter the query to split the composite value returned by AllCoords into a set of individual coordinates by using the SPLIT BY clause. Importing or linking a geom created by UnionAll creates one area. If none of the argument areas overlap, the area returned by UnionAll is the same as that returned by AllBranches.

**Split Functions**

Queries can use the Branches, Coords, and Islands split functions to split Geom columns. See the SPLIT BY clause topic for details.

- **Branches** outputs each branch of an input geometric object as a separate object.
- **Coords** outputs each coordinate of an input geometric object as a separate point object.
- **Islands** outputs each island of an input area as a separate area.

**Remarks**

- **Geom** arguments are either geometric objects of type Geometry or are object IDs.

  All spatial computations are done using the location precision parameter taken from the drawing of the first argument if it is an object ID, or the maximum possible precision if it is a geometric object.

  All spatial computations are done using the coordinate system of the first argument. Performing spatial computations using data in different coordinate systems will convert the data in the second and any subsequent arguments to the coordinate system of the first argument.

  Some functions, such as Area or Distance, allow specifying measurement units. When specified, a unit should be compatible with the coordinate system of the first argument, in that if the coordinate system is Latitude / Longitude the unit must be angular (degrees or other angular unit), and if the coordinate system is not Latitude / Longitude the unit must be metric.

  If the unit argument is omitted the functions will use the units specified in the coordinate system of the first argument adjusted with the values of the local scale parameters, the adjusted result being equivalent to the component's native unit. See the Appendices - Tables - Units topic for a list of standard units.

  When using buffer functions, the buffer distance used must be at least four times the size of the location precision factor. If precision has been set to 10 meters then the buffer distance must be larger than 40 meters.

**Examples**

See the Sql.map sample project in the Manifold CD's Examples\Help folder for a Manifold project that contains SQL examples like those below.

We have a drawing D and want to compute its bounding box. We can do this with the following query:

```
SELECT BoundingBox(AllCoords(Geom([ID]))) FROM [D];
```

We have a drawing D that contains road segments with each segment assigned a road name in a column called Name. We want to join segments for each road into a single line object. We can do this with the following query:
SELECT AllBranches(Geom([ID])), [Name] FROM [D] GROUP BY [Name];

For the examples below, suppose we have two drawings called "Restaurants" and "Metro Stations" that contain point objects, a "Roads" drawing that contains line objects, and a "Parks" drawing that contains area objects.

This example uses the \texttt{Area} function to select the two largest parks:

$$\text{SELECT TOP 2 [Park Name] FROM [Parks] ORDER BY Area([Parks].[ID]) DESC;}$$

This example uses the \texttt{Distance} function to select the nearest restaurant to each metro station:

$$\text{SELECT [Metro Station Name], (SELECT TOP 1 [Restaurant Name] FROM [Restaurants] ORDER BY Distance([Metro Stations].[ID], [Restaurants].[ID]) FROM [Metro Stations]);}$$

This example uses the \texttt{Contains} function to select all restaurants within parks:

$$\text{SELECT [Restaurant Name], [Park Name] FROM [Restaurants], [Parks] WHERE Contains([Parks].[ID], [Restaurants].[ID]);}$$

This example uses the \texttt{Intersects} function to select all roads that intersect parks:

$$\text{SELECT [Road Name], [Park Name] FROM [Roads], [Parks] WHERE Intersects([Roads].[ID], [Parks].[ID]);}$$

This example uses the \texttt{Intersects} function to select all roads that intersect other roads:

$$\text{SELECT [Roads].[Road Name], [Copy].[Road Name] FROM [Roads], [Roads] AS [Copy] WHERE [Roads].[ID] <> [Copy].[ID] AND Intersects([Roads].[ID], [Copy].[ID]);}$$

This example uses the \texttt{Adjacent} function to select all roads near a park that are close to a metro station:

$$\text{SELECT [Road Name] FROM [Roads] WHERE EXISTS (SELECT * FROM [Parks] WHERE Adjacent([Parks].[ID], [Roads].[ID])) AND EXISTS (SELECT * FROM [Metro Stations] WHERE Distance([Metro Stations].[ID], [Roads].[ID], "yd") < 200);}$$

Suppose we have two drawings containing points where each point represents a ship. Each point has a "Ship ID" column that specifies the identifying number of the ship. Drawing "Monday" shows the positions of the ships on Monday and drawing "Tuesday" shows the positions of the ships on Tuesday. We would like to create a table that shows the distance between the Monday position and the Tuesday position for each ship.

$$\text{SELECT [Monday].*, Distance([Monday].[ID], [Tuesday].[ID], "km") AS [Sail Distance] FROM [Monday] INNER JOIN [Tuesday] ON [Monday].[Ship ID] = [Tuesday].[Ship ID];}$$

This will create a table with all columns from the "Monday" drawing plus a computed column called "Sail Distance" containing the distance in kilometers between ship positions on Monday and Tuesday. \texttt{Monday} must be a projected drawing to use the optional "km" specification of units. The objects in the \texttt{Tuesday} drawing will be reprojected on the fly, if necessary, into the coordinate system of the \texttt{Monday} drawing so that the same units can be used in both for the purposes of the query.

\textbf{Advanced Example}

This example shows three queries that do the same thing with greatly different performance. This illustrates the importance of considering different SQL approaches to achieve performance gains. The queries operate on a
project containing a surface, \(S\), and a drawing, \(D\). The drawing has lines in it, and our task is to select all pixels in the surface that are in a buffer zone within 20 units of a line in the drawing.

**Query 1** below is very slow because it re-computes the buffer returned by the inner SELECT for each pixel. The Manifold query engine is smart enough to cache the selected lines the SELECT operates upon, but it is not smart enough to cache the result of the aggregate on these lines. The solution is to get rid of computations in the column list of the SELECT (as is done in **Query 2**), or, better yet, to get rid of the SELECT in favor of a JOIN (as is done in **Query 3**).

The number of seconds cited for query execution of each query use a sample set of data that provides a reasonable comparison of the efficiency of the three queries.

**Query 1** - Approximately 2500 seconds:

```sql
UPDATE [S] SET [Selection (I)] = True
WHERE Intersects(SELECT Buffer(AllBranches([ID]), 20) FROM [D],
    NewPointLatLon([S].[Longitude (I)], [S].[Latitude (I)]))
```

**Query 2** - Computations in the column list are reduced to a minimum - 29 seconds:

```sql
UPDATE [S] SET [Selection (I)] = True
WHERE (SELECT Min(Distance([ID], NewPointLatLon([S].[Longitude (I)], [S].[Latitude (I)]))) FROM [D]) <= 20
```

**Query 3** - SELECT is replaced with an INNER JOIN - 3 seconds:

```sql
UPDATE (SELECT [S].[Selection (I)] FROM [S] INNER JOIN [D] ON Distance([D].[ID], NewPointLatLon([S].[Longitude (I)], [S].[Latitude (I)])) <= 20) SET [Selection (I)] = True
```

Reducing the time required for the query from 2500 to 3 seconds is a huge gain, certainly worth the time spent learning constructs such as INNER JOIN.

**Examples of Queries using Geometry Data**

See the Linked Drawings topic and the Geometry in Tables topic.

**See Also**

- Selecting Objects with Queries
- Using SQL to Select Map Objects
- Geocoding Extensions
- Raster Extensions
- Linked Drawings
- Geometry in Tables
- Queries and Geoms
Geocoding Extensions

Geocoding extensions operate with Manifold's geocoding engine to perform spatial operations based upon an address string or zip code.

Geocoding extensions will not work unless the optional Geocoding Tools package has been installed and also an appropriate geocoding database has been installed. See the Street Address Geocoding topic for information on geocoding databases that may be used.

Geocoding extensions using the Manifold US streets geocoding database will not work with Manifold IMS unless the geocoding database is installed within the Manifold application installation folder as specified in the Tools - Options file locations settings and as also specified within the IMS config.txt settings.

The following table lists available geocoding functions:

<table>
<thead>
<tr>
<th>Function</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>CloseToAddress(geom, address, distance[, unit])</td>
<td>Returns True if the object lies within the specified distance of the address; otherwise returns False.</td>
</tr>
<tr>
<td>CloseToZip(geom, zip, distance[, unit])</td>
<td>Returns True if the object lies within the specified distance of the zip code centroid; otherwise returns False.</td>
</tr>
<tr>
<td>DistanceToAddress(geom, address[, unit])</td>
<td>Returns the distance between the object and the address in the specified units.</td>
</tr>
<tr>
<td>DistanceToZip(geom, zip[, unit])</td>
<td>Returns the distance between the object and the address in the specified units.</td>
</tr>
<tr>
<td>LocateAddress(address)</td>
<td>Returns the location of the address as a point.</td>
</tr>
<tr>
<td>LocateAddressLat(address)</td>
<td>Returns the latitude of the address.</td>
</tr>
<tr>
<td>LocateAddressLon(address)</td>
<td>Returns the longitude of the address.</td>
</tr>
<tr>
<td>LocateZip(zip)</td>
<td>Returns the location of the zip code centroid as a point.</td>
</tr>
<tr>
<td>LocateZipLat(zip)</td>
<td>Returns the latitude of the zip code centroid. Works only with US zip codes.</td>
</tr>
<tr>
<td>LocateZipLon(zip)</td>
<td>Returns the longitude of the zip code centroid. Works only with US zip codes.</td>
</tr>
</tbody>
</table>

Remarks

Geom arguments are either geometric objects of type Geometry or are object IDs. The address and zip arguments are strings.

If an object is a line or an area the object's centroid is used for distance calculations.

If an address string produces more than one match, the system automatically selects the closest of the building-level matches (possibly with an "unknown street name, possible misspelling" error). If an address string produces no building-level matches, CloseToAddress returns False, and DistanceToAddress, LocateAddress, LocateAddressLat and LocateAddressLon return a null value.

If a zip code string is invalid (no matches in the geocoding database) CloseToZip returns False, and DistanceToZip, LocateZip, LocateZipLat and LocateZipLon return a null value.

If the optional distance unit is omitted, the system will use the native measurement unit of the drawing or meters if the drawing is not projected. See the Appendices - Tables - Units topic for a list of standard units.

Distances are great circle distances computed over a WGS84 ellipsoid and are accurate to 1 meter.
The geocoding functions cache returned geocoding data between subsequent calls. This means that a \texttt{LocateAddressLat} followed by a \texttt{LocateAddressLon} runs a geocoder search only once on the address.

Functions can be used from IMS as long as the geocoding database is located within the Manifold application installation folder (usually \texttt{C:\Program Files\Manifold System}). To keep the user interface simple and to avoid the complication of dealing with possible user errors when entering address information into forms, many web applications with IMS will use the \texttt{CloseToZip} or \texttt{DistanceToZip} functions since these require the user to merely enter the ZIP code correctly. For many applications, such as locating a dealer, finding the closest objects to the ZIP centroid provides acceptable accuracy.

**Examples**

For the examples below, suppose we have two drawings called "Clients" and "Dealers" that contain point objects and the "Clients" drawing that contains a zip code column.

This example uses the \texttt{CloseToAddress} function to locate all dealers within 10 miles of 330 Lytton Ave, Palo Alto:

\begin{verbatim}
SELECT * FROM [Dealers] WHERE CloseToAddress([ID], "330 Lytton Ave, Palo Alto, CA, 94301", 10, "mi");
\end{verbatim}

This example uses the \texttt{CloseToZip} function to select all dealers within 10 miles of the centroid for the 89701 zip code:

\begin{verbatim}
SELECT * FROM [Dealers] WHERE CloseToZip([ID], "89701", 10, "mi");
\end{verbatim}

This example uses the \texttt{DistanceToZip} function to select all dealers within 5 to 15 miles of the centroid for the 89701 zip code:

\begin{verbatim}
SELECT * FROM [Dealers] WHERE DistanceToZip([ID], "89701", "mi") BETWEEN 5 AND 15;
\end{verbatim}

This example uses the \texttt{LocateZipLat} and \texttt{LocateZipLon} functions to output a lat/lon location for the zip code centroid of each client:

\begin{verbatim}
SELECT [Name], LocateZipLat([Zip]) AS [Latitude], LocateZipLon([Zip]) AS [Longitude] FROM [Clients];
\end{verbatim}

**See Also**

Geocoding Tools
Street Address Geocoding
Selecting Objects with Queries
Using SQL to Select Map Objects
Spatial Extensions
Raster Extensions
## Raster Extensions

A raster data set is one in which data is stored as pixels arranged in rows of specified width. Images and surfaces are raster data sets within Manifold System. Manifold SQL provides functions for analyzing and manipulating raster data in images and surfaces. Note that some functions are available only if the optional Surface Tools extension is installed.

The following table lists available raster functions. Functions are often employed with data in tables of type `Geometry`, `Geometry (SHP)` or `Geometry (WKB)`. See the Geometry in Tables and the Queries and Geoms topics for additional information on storing and using geometry in tables.

In the following functions, `p` is either an image or surface (a raster data set consisting of pixels). `s` is a surface. `i` is an image. `g` is a geom.

<table>
<thead>
<tr>
<th>Function</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>AspectHeight(s, g)</td>
<td>Returns the aspect of s at the specified location. The location is either the centroid of a geom or a pair of XY coordinates. Surface Tools only.</td>
</tr>
<tr>
<td>AspectHeight(s, x, y)</td>
<td></td>
</tr>
<tr>
<td>AspectHeight(s, x, y, w)</td>
<td>Passing the w parameter computes aspect over a window of specified size (1 for 3x3, 2 for 5x5, etc).</td>
</tr>
<tr>
<td>Blur(p, g)</td>
<td>Returns the value of the Blur filter at the specified location. The location is either the centroid of a geom or a pair of XY coordinates.</td>
</tr>
<tr>
<td>Blur(p, x, y)</td>
<td></td>
</tr>
<tr>
<td>BlurHeight(s, g)</td>
<td></td>
</tr>
<tr>
<td>BlurHeight(s, x, y)</td>
<td></td>
</tr>
<tr>
<td>DifferenceE(p, g)</td>
<td>Returns the value of the DifferenceE filter at the specified location. The location is either the centroid of a geom or a pair of XY coordinates.</td>
</tr>
<tr>
<td>DifferenceE(p, x, y)</td>
<td></td>
</tr>
<tr>
<td>DifferenceEHeight(s, g)</td>
<td></td>
</tr>
<tr>
<td>DifferenceEHeight(s, x, y)</td>
<td></td>
</tr>
<tr>
<td>DifferenceN(p, g)</td>
<td>Returns the value of the DifferenceN filter at the specified location. The location is either the centroid of a geom or a pair of XY coordinates.</td>
</tr>
<tr>
<td>DifferenceN(p, x, y)</td>
<td></td>
</tr>
<tr>
<td>DifferenceNHeight(s, g)</td>
<td></td>
</tr>
<tr>
<td>DifferenceNHeight(s, x, y)</td>
<td></td>
</tr>
<tr>
<td>DifferenceNE(p, g)</td>
<td>Returns the value of the DifferenceNE filter at the specified location. The location is either the centroid of a geom or a pair of XY coordinates.</td>
</tr>
<tr>
<td>DifferenceNE(p, x, y)</td>
<td></td>
</tr>
<tr>
<td>DifferenceNEHeight(s, g)</td>
<td></td>
</tr>
<tr>
<td>DifferenceNEHeight(s, x, y)</td>
<td></td>
</tr>
<tr>
<td>DifferenceNW(p, g)</td>
<td>Returns the value of the DifferenceNW filter at the specified location. The location is either the centroid of a geom or a pair of XY coordinates.</td>
</tr>
<tr>
<td>DifferenceNW(p, x, y)</td>
<td></td>
</tr>
<tr>
<td>DifferenceNWHeight(s, g)</td>
<td></td>
</tr>
<tr>
<td>DifferenceNWHeight(s, x, y)</td>
<td></td>
</tr>
<tr>
<td>DifferenceS(p, g)</td>
<td>Returns the value of the DifferenceS filter at the specified location. The location is either the centroid of a geom or a pair of XY coordinates.</td>
</tr>
<tr>
<td>DifferenceS(p, x, y)</td>
<td></td>
</tr>
<tr>
<td>DifferenceSHeight(s, g)</td>
<td></td>
</tr>
<tr>
<td>DifferenceSHeight(s, x, y)</td>
<td></td>
</tr>
<tr>
<td>DifferenceSE(p, g)</td>
<td>Returns the value of the DifferenceSE filter at the specified location. The location is either the centroid of a geom or a pair of XY coordinates.</td>
</tr>
<tr>
<td>DifferenceSE(p, x, y)</td>
<td></td>
</tr>
<tr>
<td>DifferenceSEHeight(s, g)</td>
<td></td>
</tr>
<tr>
<td>DifferenceSEHeight(s, x, y)</td>
<td></td>
</tr>
</tbody>
</table>
DifferenceSW(p, g)
- Returns the value of the DifferenceSW filter at the specified location. The location is either the centroid of a geom or a pair of XY coordinates.

DifferenceSW(p, x, y)

DifferenceSWHeight(s, g)

DifferenceSWHeight(s, x, y)

DifferenceW(p, g)
- Returns the value of the DifferenceW filter at the specified location. The location is either the centroid of a geom or a pair of XY coordinates.

DifferenceW(p, x, y)

DifferenceWHeight(s, g)

DifferenceWHeight(s, x, y)

Gravity(drawing, column, neighbors, g)
- Returns the value of the surface interpolated using gravity interpolation method at the specified location. The location is either the centroid of a geom or a pair of XY coordinates. The surface is built from the specified drawing using the specified column as the height. Setting the number of neighbors to -1 will use the default number of neighbors. Surface Tools only.

Gravity(drawing, column, neighbors, x, y)

HasPixel(p, g)
- Returns True if s has a visible pixel at the specified location and False otherwise. The location is either the centroid of a geom or a pair of XY coordinates.

HasPixel(p, x, y)

Height(s, g)
- Returns average height of pixels covered by a geom or height of a pixel at an XY location in the given surface.

Height(s, x, y)

HeightAvg(s, g)
- Returns average height of pixels covered by a geom or average height of pixels in a window of specified size (1 for 3x3, 2 for 5x5, etc) centered at the specified location. The location is either the centroid of a geom or a pair of XY coordinates. Window computations are available in Surface Tools only.

HeightAvg(s, g, w)

HeightAvg(s, x, y, w)

HeightCurvMean(s, g)
- Returns the mean curvature of s at the specified location. The location is either the centroid of a geom or a pair of XY coordinates. Passing the w parameter computes curvature over a window of specified size (1 for 3x3, 2 for 5x5, etc). Surface Tools only.

HeightCurvMean(s, x, y)

HeightCurvMean(s, x, y, w)

HeightCurvPlan(s, g)
- Returns the plan curvature of s at the specified location. The location is either the centroid of a geom or a pair of XY coordinates. Passing the w parameter computes curvature over a window of specified size (1 for 3x3, 2 for 5x5, etc). Surface Tools only.

HeightCurvPlan(s, x, y)

HeightCurvPlan(s, x, y, w)

HeightCurvProfile(s, g)
- Returns the profile curvature of s at the specified location. The location is either the centroid of a geom or a pair of XY coordinates. Passing the w parameter computes curvature over a window of specified size (1 for 3x3, 2 for 5x5, etc). Surface Tools only.

HeightCurvProfile(s, x, y)

HeightCurvProfile(s, x, y, w)

HeightDiversity(s, g)
- Returns the number of different height values in pixels covered by a geom or the number of different height values in pixels in a window of specified size (1 for 3x3, 2 for 5x5, etc) centered at the specified location. The location is either the centroid of a geom or a pair of XY coordinates. Window computations are available in Surface Tools only.

HeightDiversity(s, g, w)

HeightDiversity(s, x, y, w)

HeightDiversityIndex(s, g)
- Returns the diversity index of height values in pixels covered by a geom or the diversity index of height values in pixels in a window of specified size (1 for 3x3, 2 for 5x5, etc) centered at the specified location. The location is either the centroid of a geom or a pair of XY coordinates. The diversity index is a value from 0 to 1. The greater the value of the diversity index, the less similar the height values. Window computations are available in Surface Tools only.

HeightDiversityIndex(s, g, w)

HeightDiversityIndex(s, x, y, w)

HeightMaj(s, g)
- Returns the most frequently occurring height of pixels covered by a geom or the most frequently occurring height
HeightMaj(s, g, w)  \-
| of pixels in a window of specified size (1 for 3x3, 2 for 5x5, etc) centered at the specified location. The location is either the centroid of a geom or a pair of XY coordinates. Window computations are available in Surface Tools only.

HeightMaj(s, x, y, w)  \-
| of pixels in a window of specified size (1 for 3x3, 2 for 5x5, etc) centered at the specified location. The location is either the centroid of a geom or a pair of XY coordinates. Window computations are available in Surface Tools only.

HeightMax(s)  \-
| Returns maximum height of pixels covered by a geom or maximum height of pixels in a window of specified size (1 for 3x3, 2 for 5x5, etc) centered at the specified location or maximum height of all pixels. The location is either the centroid of a geom or a pair of XY coordinates. Window computations are available in Surface Tools only.

HeightMax(s, g)  \-
| Returns maximum height of pixels covered by a geom or maximum height of pixels in a window of specified size (1 for 3x3, 2 for 5x5, etc) centered at the specified location or maximum height of all pixels. The location is either the centroid of a geom or a pair of XY coordinates. Window computations are available in Surface Tools only.

HeightMax(s, g, w)  \-
| Returns maximum height of pixels covered by a geom or maximum height of pixels in a window of specified size (1 for 3x3, 2 for 5x5, etc) centered at the specified location or maximum height of all pixels. The location is either the centroid of a geom or a pair of XY coordinates. Window computations are available in Surface Tools only.

HeightMax(s, x, y, w)  \-
| Returns maximum height of pixels covered by a geom or maximum height of pixels in a window of specified size (1 for 3x3, 2 for 5x5, etc) centered at the specified location or maximum height of all pixels. The location is either the centroid of a geom or a pair of XY coordinates. Window computations are available in Surface Tools only.

HeightMed(s, g)  \-
| Returns median (mean) height of pixels covered by a geom or median height of pixels in a window of specified size (1 for 3x3, 2 for 5x5, etc) centered at the specified location. The location is either the centroid of a geom or a pair of XY coordinates. Window computations are available in Surface Tools only.

HeightMed(s, g, w)  \-
| Returns median (mean) height of pixels covered by a geom or median height of pixels in a window of specified size (1 for 3x3, 2 for 5x5, etc) centered at the specified location. The location is either the centroid of a geom or a pair of XY coordinates. Window computations are available in Surface Tools only.

HeightMed(s, x, y, w)  \-
| Returns median (mean) height of pixels covered by a geom or median height of pixels in a window of specified size (1 for 3x3, 2 for 5x5, etc) centered at the specified location. The location is either the centroid of a geom or a pair of XY coordinates. Window computations are available in Surface Tools only.

HeightMin(s)  \-
| Returns minimum height of pixels covered by a geom or minimum height of pixels in a window of specified size (1 for 3x3, 2 for 5x5, etc) centered at the specified location or minimum height of all pixels. The location is either the centroid of a geom or a pair of XY coordinates. Window computations are available in Surface Tools only.

HeightMin(s, g)  \-
| Returns minimum height of pixels covered by a geom or minimum height of pixels in a window of specified size (1 for 3x3, 2 for 5x5, etc) centered at the specified location or minimum height of all pixels. The location is either the centroid of a geom or a pair of XY coordinates. Window computations are available in Surface Tools only.

HeightMin(s, g, w)  \-
| Returns minimum height of pixels covered by a geom or minimum height of pixels in a window of specified size (1 for 3x3, 2 for 5x5, etc) centered at the specified location or minimum height of all pixels. The location is either the centroid of a geom or a pair of XY coordinates. Window computations are available in Surface Tools only.

HeightMin(s, x, y, w)  \-
| Returns minimum height of pixels covered by a geom or minimum height of pixels in a window of specified size (1 for 3x3, 2 for 5x5, etc) centered at the specified location or minimum height of all pixels. The location is either the centroid of a geom or a pair of XY coordinates. Window computations are available in Surface Tools only.

HeightSum(s)  \-
| Returns summed height of pixels covered by a geom or summed height of pixels in a window of specified size (1 for 3x3, 2 for 5x5, etc) centered at the specified location. The location is either the centroid of a geom or a pair of XY coordinates. Used to compute cumulative figures such as population in a given region or to compute the volume of a surface (the HeightSum value multiplied by the size of a pixel). Window computations are available in Surface Tools only.

HeightSum(s, g)  \-
| Returns summed height of pixels covered by a geom or summed height of pixels in a window of specified size (1 for 3x3, 2 for 5x5, etc) centered at the specified location. The location is either the centroid of a geom or a pair of XY coordinates. Used to compute cumulative figures such as population in a given region or to compute the volume of a surface (the HeightSum value multiplied by the size of a pixel). Window computations are available in Surface Tools only.

HeightSum(s, g, w)  \-
| Returns summed height of pixels covered by a geom or summed height of pixels in a window of specified size (1 for 3x3, 2 for 5x5, etc) centered at the specified location. The location is either the centroid of a geom or a pair of XY coordinates. Used to compute cumulative figures such as population in a given region or to compute the volume of a surface (the HeightSum value multiplied by the size of a pixel). Window computations are available in Surface Tools only.

HeightSum(s, x, y, w)  \-
| Returns summed height of pixels covered by a geom or summed height of pixels in a window of specified size (1 for 3x3, 2 for 5x5, etc) centered at the specified location. The location is either the centroid of a geom or a pair of XY coordinates. Used to compute cumulative figures such as population in a given region or to compute the volume of a surface (the HeightSum value multiplied by the size of a pixel). Window computations are available in Surface Tools only.

HighPass1(p, g)  \-
| Returns the value of the HighPass1 filter at the specified location. The location is either the centroid of a geom or a pair of XY coordinates.

HighPass1(p, x, y)  \-
| Returns the value of the HighPass1 filter at the specified location. The location is either the centroid of a geom or a pair of XY coordinates.

HighPass1Height(s, g)  \-
| Returns the value of the HighPass1 filter at the specified location. The location is either the centroid of a geom or a pair of XY coordinates.

HighPass1Height(s, x, y)  \-
| Returns the value of the HighPass1 filter at the specified location. The location is either the centroid of a geom or a pair of XY coordinates.

HighPass2(p, g)  \-
| Returns the value of the HighPass2 filter at the specified location. The location is either the centroid of a geom or a pair of XY coordinates.

HighPass2(p, x, y)  \-
| Returns the value of the HighPass2 filter at the specified location. The location is either the centroid of a geom or a pair of XY coordinates.

HighPass2Height(s, g)  \-
| Returns the value of the HighPass2 filter at the specified location. The location is either the centroid of a geom or a pair of XY coordinates.

HighPass2Height(s, x, y)  \-
| Returns the value of the HighPass2 filter at the specified location. The location is either the centroid of a geom or a pair of XY coordinates.

HighPass3(p, g)  \-
| Returns the value of the HighPass3 filter at the specified location. The location is either the centroid of a geom or a pair of XY coordinates.

HighPass3(p, x, y)  \-
| Returns the value of the HighPass3 filter at the specified location. The location is either the centroid of a geom or a pair of XY coordinates.

HighPass3Height(s, g)  \-
| Returns the value of the HighPass3 filter at the specified location. The location is either the centroid of a geom or a pair of XY coordinates.

HighPass3Height(s, x, y)  \-
| Returns the value of the HighPass3 filter at the specified location. The location is either the centroid of a geom or a pair of XY coordinates.

InterpolateHeight(s, g, radius)  \-
| Returns the value of s at the specified location if it is visible and the interpolated value if it is invisible. The location is either the centroid of a geom or a pair of XY coordinates. The interpolated value is computed as the average value of s in the smallest box centered at the current pixel that contains at least one visible pixel. If all pixels in s are invisible, the function returns Null. The last parameter specifies an interpolation radius over which interpolation occurs, a negative value specifies infinite radius. See the discussion for the Interpolate (Parameter) transform operator.

InterpolateHeight(s, x, y, radius)  \-
| Returns the value of s at the specified location if it is visible and the interpolated value if it is invisible. The location is either the centroid of a geom or a pair of XY coordinates. The interpolated value is computed as the average value of s in the smallest box centered at the current pixel that contains at least one visible pixel. If all pixels in s are invisible, the function returns Null. The last parameter specifies an interpolation radius over which interpolation occurs, a negative value specifies infinite radius. See the discussion for the Interpolate (Parameter) transform operator.

InterpolateRowHeight(s, g)  \-
| Returns the value of s at the specified location if it is visible and the interpolated value if it is invisible. The location is either the centroid of a geom or a pair of XY coordinates. The interpolated value is computed as the value of the straight line segment between the nearest visible pixels in the same row of s. If all pixels in the current row of s are
invisible, the function returns Null. Overall, this function works much faster than Interpolate but is also less accurate.

Kriging(drawing, column, neighbors, g)
Kriging(drawing, column, neighbors, x, y)
Kriging(drawing, column, neighbors, model, voronoi, g)
Kriging(drawing, column, neighbors, model, voronoi, x, y)

Returns the value of the interpolated surface at the specified location. The location is either the centroid of a geom or a pair of XY coordinates. The surface is built from the specified drawing using the specified column as the height. Line objects participate as sets of their coordinates. Area objects are ignored. Setting the number of neighbors to -1 will use the default number of neighbors. The model argument is a case-insensitive string, e.g., "spherical" or "gaussian". Some models are only available in the Surface Tools extension. The voronoi argument is a boolean (if true, only Voronoi neighbors are used).

KrigingMedianPolish(drawing, column, neighbors, g)
KrigingMedianPolish(drawing, column, neighbors, x, y)
KrigingMedianPolish(drawing, column, neighbors, model, voronoi, g)
KrigingMedianPolish(drawing, column, neighbors, model, voronoi, x, y)

Returns the value of the surface interpolated using the median-polish kriging interpolation method at the specified location. The location is either the centroid of a geom or a pair of XY coordinates. The surface is built from the specified drawing using the specified column as the height. Line objects participate as sets of their coordinates. Area objects are ignored. Setting the number of neighbors to -1 will use the default number of neighbors. The model argument is a case-insensitive string, e.g., "spherical" or "gaussian". The voronoi argument is a boolean (if true, only Voronoi neighbors are used). This function requires the Surface Tools extension.

Laplace1(p, g)
Laplace1(p, x, y)
Laplace1Height(s, g)
Laplace1Height(s, x, y)

Returns the value of the Laplace1 filter at the specified location. The location is either the centroid of a geom or a pair of XY coordinates.

Laplace2(p, g)
Laplace2(p, x, y)
Laplace2Height(s, g)
Laplace2Height(s, x, y)

Returns the value of the Laplace2 filter at the specified location. The location is either the centroid of a geom or a pair of XY coordinates.

LowPass1(p, g)
LowPass1(p, x, y)
LowPass1Height(s, g)
LowPass1Height(s, x, y)

Returns the value of the LowPass1 filter at the specified location. The location is either the centroid of a geom or a pair of XY coordinates.

LowPass2(p, g)
LowPass2(p, x, y)
LowPass2Height(s, g)
LowPass2Height(s, x, y)

Returns the value of the LowPass2 filter at the specified location. The location is either the centroid of a geom or a pair of XY coordinates.

LowPass3(p, g)
LowPass3(p, x, y)
LowPass3Height(s, g)
LowPass3Height(s, x, y)

Returns the value of the LowPass3 filter at the specified location. The location is either the centroid of a geom or a pair of XY coordinates.

MedianCross(p, g)
MedianCross(p, x, y)
MedianCrossHeight(s, g)
MedianCrossHeight(s, x, y)

Returns the value of the MedianCross filter at the specified location. The location is either the centroid of a geom or a pair of XY coordinates.

MedianSquare(p, g)

Returns the value of the MedianSquare filter at the specified location. The location is either the centroid of a
MedianSquare(p, x, y) Returns the value of the MedianSquare filter at the specified location. The location is either the centroid of a geom or a pair of XY coordinates.

MedianSquareHeight(s, g) Returns the area of a single pixel in p in the specified units, or in the units of p.

MedianSquareHeight(s, x, y) Returns the color of a pixel at the specified location. The location is either the centroid of a geom or a pair of XY coordinates.

MedianSquare5Height(s, g) Returns the height of a single pixel in p in the specified units, or in the units of p.

MedianSquare5Height(s, x, y) Returns the number of columns in p.

MedianSquare5Height(s, x, y) Returns the number of rows in p.

MedianSquare5Height(s, x, y) Returns the value of the MedianSquare5 filter at the specified location. The location is either the centroid of a geom or a pair of XY coordinates.

PixelArea(p[, unit]) Returns the value of the Sharpen filter at the specified location. The location is either the centroid of a geom or a pair of XY coordinates.

Pixel(p, g) Returns the value of the Sharpen filter at the specified location. The location is either the centroid of a geom or a pair of XY coordinates.

Pixel(p, x, y) Returns the value of the SharpenMore filter at the specified location. The location is either the centroid of a geom or a pair of XY coordinates.

PixelHeight(p[, unit]) Returns the slope of s at the specified location. The location is either the centroid of a geom or a pair of XY coordinates. Passing the w parameter computes slope over a window of specified size (1 for 3x3, 2 for 5x5, etc).

PixelHeight(p, x, y) Returns the slope of s at the specified location. The location is either the centroid of a geom or a pair of XY coordinates.

PixelHeight(s, x, y) Returns the slope of s at the specified location. The location is either the centroid of a geom or a pair of XY coordinates.

PixelWidth(p[, unit]) Returns the slope of s at the specified location. The location is either the centroid of a geom or a pair of XY coordinates.

Sharpen(p, g) Returns the value of the Sharpen filter at the specified location. The location is either the centroid of a geom or a pair of XY coordinates.

Sharpen(p, x, y) Returns the value of the Sharpen filter at the specified location. The location is either the centroid of a geom or a pair of XY coordinates.

SharpenHeight(s, g) Returns the value of the SharpenMore filter at the specified location. The location is either the centroid of a geom or a pair of XY coordinates.

SharpenHeight(s, x, y) Returns the value of the SharpenMore filter at the specified location. The location is either the centroid of a geom or a pair of XY coordinates.

SharpenMore(p, g) Returns the value of the SharpenMore filter at the specified location. The location is either the centroid of a geom or a pair of XY coordinates.

SharpenMore(p, x, y) Returns the value of the SharpenMore filter at the specified location. The location is either the centroid of a geom or a pair of XY coordinates.

SharpenMoreHeight(s, g) Returns the value of the SharpenMore filter at the specified location. The location is either the centroid of a geom or a pair of XY coordinates.

SharpenMoreHeight(s, x, y) Returns the value of the SharpenMore filter at the specified location. The location is either the centroid of a geom or a pair of XY coordinates.

SlopeHeight(s, g) (was: Slope) Returns the value of the Tile filter at the specified location. The location is either the centroid of a geom or a pair of XY coordinates. The last parameter is the tile size.

SlopeHeight(s, x, y) Returns the value of the Tile filter at the specified location. The location is either the centroid of a geom or a pair of XY coordinates. The last parameter is the tile size.

SlopeHeight(s, x, y) Returns the value of the Tile filter at the specified location. The location is either the centroid of a geom or a pair of XY coordinates. The last parameter is the tile size.

SlopeHeight(s, x, y) Returns the value of the Tile filter at the specified location. The location is either the centroid of a geom or a pair of XY coordinates. The last parameter is the tile size.

Surface Tools only.

Tile(p, g, n) Returns the value of the Tile filter at the specified location. The location is either the centroid of a geom or a pair of XY coordinates. The last parameter is the tile size.

Tile(p, x, y, n) Returns the value of the Tile filter at the specified location. The location is either the centroid of a geom or a pair of XY coordinates. The last parameter is the tile size.

TileHeight(s, g, n) Returns the value of the TileMedian filter at the specified location. The location is either the centroid of a geom or a pair of XY coordinates. The last parameter is the tile size.

TileHeight(s, x, y, n) Returns the value of the TileMedian filter at the specified location. The location is either the centroid of a geom or a pair of XY coordinates. The last parameter is the tile size.

TileMedian(p, g, n) Returns the value of the TileMedian filter at the specified location. The location is either the centroid of a geom or a pair of XY coordinates. The last parameter is the tile size.

TileMedian(p, x, y, n) Returns the value of the TileMedian filter at the specified location. The location is either the centroid of a geom or a pair of XY coordinates. The last parameter is the tile size.

TileMedianHeight(s, g, n) Returns the value of the TileMedian filter at the specified location. The location is either the centroid of a geom or a pair of XY coordinates. The last parameter is the tile size.

TileMedianHeight(s, x, y, n) Returns the value of the TileMedian filter at the specified location. The location is either the centroid of a geom or a pair of XY coordinates. The last parameter is the tile size.

Triangulation(drawing, column, corners, g) Returns the value of the surface interpolated using the triangulation interpolation method at the specified location. The location is either the centroid of a geom or a pair of XY coordinates. The surface is built from the specified drawing using the specified column as the height. Line objects participate as sets of their coordinates. Area objects are ignored. The corners argument specifies the height to use for corners, and can be set to NULL.

Triangulation(drawing, column, corners, x, y) Returns the value of the surface interpolated using the contour triangulation interpolation method at the specified location. The location is either the centroid of a geom or a pair of XY coordinates. The surface is built from the specified drawing using the specified column as the height. Line objects participate as sets of their coordinates. Area objects are ignored. The corners argument specifies the height to use for corners, and can be set to NULL.

TriangulationContours(drawing, column,
The `TriangulationContours` function builds a surface from a specified drawing using a specified column as the height. Line objects participate as sets of their segments, with intersections between segments being resolved automatically. Area objects are ignored. The `corners` argument specifies the height to use for corners, and can be set to NULL.

The `Volume` function returns the volume covered by a geom or the volume of pixels in a window of specified size (1 for 3x3, 2 for 5x5, etc) centered at the specified location. The location is either the centroid of a geom or a pair of XY coordinates. The function assumes surface heights are in meters, and returns the volume in cubic meters. To compute volume, the system takes all pixels, computes the volume of each pixel multiplying its height (which can be negative) by its area, and sums the resulting volumes. Hence, the volume can be less than zero. Window computations are available in Surface Tools only.

### Color Manipulation Functions

Colors within images may be manipulated using the following functions. In the following functions, `c` is a color.

<table>
<thead>
<tr>
<th>Function</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARGB(a, r, g, b)</td>
<td>Combines given values for alpha, red, green and blue channels into a color.</td>
</tr>
<tr>
<td>BGR(b, g, r)</td>
<td>Combines given values for blue, green and red channels into a color.</td>
</tr>
<tr>
<td>BGRA(b, g, r, a)</td>
<td>Combines the given values for blue, green, red and alpha channels into a color.</td>
</tr>
<tr>
<td>Channel(c, n)</td>
<td>Extracts the value of the given channel of the color where n specifies the channel: 0 = blue, 1 = green, 2 = red, 3 = alpha.</td>
</tr>
<tr>
<td>HLS(h, l, s)</td>
<td>Combines given values for hue, lightness and saturation into a color.</td>
</tr>
<tr>
<td>Hue(c)</td>
<td>Extracts the hue of a color.</td>
</tr>
<tr>
<td>Intensity(c)</td>
<td>Extracts the brightness of a color (also known as &quot;luma&quot;). The name of the function has been kept for compatibility with previous versions of Manifold.</td>
</tr>
<tr>
<td>Lightness(c)</td>
<td>Extracts the lightness of a color.</td>
</tr>
<tr>
<td>RGB(r, g, b)</td>
<td>Combines given values for red, green and blue channels into a color.</td>
</tr>
<tr>
<td>Saturation(c)</td>
<td>Extracts the saturation of a color.</td>
</tr>
</tbody>
</table>

**Lightness** in the above is the same as `L` in classic HSL color space as used in traditional graphics arts. It is computed by the following formula:

$$l = \frac{\max(r, g, b) + \min(r, g, b)}{2}$$

See the Expressions topic for additional SQL expressions.

### Example: Transferring Aspect to Points
Suppose we have a drawing of points that are located in the same region of interest as a surface and we would like to compute the aspect of the surface at each point and to save that value as a data attribute for each point. This is easy to do using SQL as follows:

1. Open the drawing's table and add a new floating-point column named **Aspect**.
2. Create a new query.
3. Open the query and enter the following text for the query (substitute the name of the drawing for "Drawing" and the name of the surface for "Surface"):

   ```sql
   UPDATE [Drawing] SET [Aspect] = AspectHeight([Surface], Centroid([Geom (I)]));
   ```

4. Run the query. This will set the values in the **Aspect** column for each point to the aspect of the surface at that point's location.

**See Also**

Spatial Extensions

[Geocoding Extensions](#)
Comparison with Jet SQL

Jet is the Microsoft database engine used to build Microsoft's Access DBMS. Because it is available to Microsoft software developers as a free redistributable many third party applications also use Jet as the database application powering the application. Since Jet includes Microsoft's implementation of SQL, Jet SQL has become a very widely distributed implementation of SQL.

Enhanced Features of Manifold SQL

Manifold SQL provides the following enhanced features compared to Jet SQL:

- More powerful joins (can include multiple columns of any type, and use any Boolean expression).
- **SKIP** quantifier.
- No restrictions on types of nested joins (can nest INNER JOIN inside, for example, a LEFT JOIN and vice versa), FULL JOIN and CROSS JOIN operators.
- No restrictions on the number of logical expressions used in the HAVING and WHERE clauses.
- The ability to use records and lists of values in comparisons, BETWEEN AND, IN, and other operators.
- **CASE**, **COALESCE**, **NULLIF** and other ANSI operators.
- **LIKE** operator and true regular expressions in queries.
- **UNIQUE** operator.
- A pivot expression used in a TRANSFORM statement is not required to return a string.
- **SPLIT BY** and **LEAVING** clauses.
- **CAST** and other ANSI functions for use in expressions.
- Spatial extensions
- Geocoding extensions
- Raster extensions

Jet SQL Features Not Supported in Manifold SQL

Manifold SQL does not support the following features of Jet SQL:

- Security statements, such as GRANT and REVOKE.
- Transaction statements, such as COMMIT and ROLLBACK.
- Procedure statements.
- Constraints and indices (both are widely used implicitly in Manifold).
- NULL values in tables. NULL values in queries are fully supported.

See Also

Comparison with ANSI SQL
Data Types
Expressions
SQL Reserved Words / Index
Comparison with ANSI SQL

Manifold SQL is generally ANSI 92 compliant. However, certain ANSI SQL features are not implemented in Manifold SQL. Conversely, Manifold SQL includes reserved words and features not supported in ANSI SQL. These differences are normally exactly the same as those encountered when comparing Microsoft Jet SQL with ANSI SQL.

Manifold SQL and ANSI SQL each have different reserved words and data types. For more information, see the SQL Reserved Words / Index and Data types topics.

Enhanced Features of Manifold SQL

Manifold SQL provides the following enhanced features compared to ANSI SQL:

- Ability to change the type or name of a column with the ALTER TABLE statement.
- TOP and SKIP quantifiers.
- Ability to use the GROUP BY and ORDER BY clauses with expressions.
- Ability to use the ORDER BY clause in subqueries.
- VBScript-like functions for use in expressions.
- LIKEX operator and true regular expressions.
- TRANSFORM statement
- Additional aggregate functions, such as StDev and VarP
- Aggregate functions using DISTINCT with expressions (e.g., SUM(DISTINCT [A]+[B])).
- SPLIT BY and LEAVING clauses.
- PARAMETERS declaration for defining parameter queries.
- Spatial extensions
- Geocoding extensions
- Raster extensions

ANSI SQL Features Not Supported in Manifold SQL

Manifold SQL does not support the following ANSI SQL features:

- Security statements, such as GRANT and LOCK.
- Transaction statements, such as COMMIT and ROLLBACK.
- Constraints and indices (both are widely used implicitly in Manifold).
- NULL values in tables. NULL values in queries are fully supported.

See Also

Comparison with Jet SQL
Data Types
Expressions
SQL Reserved Words / Index
Expressions

SQL queries can contain expressions. Expressions can contain literals and references to columns used by a query. There are several types of expressions with the most frequently used types being generic expressions, numeric expressions, string expressions, date expressions, and Boolean expressions.

Generic Expressions

Generic expressions are expressions that can return values of any type. Generic expressions can include the following functions:

<table>
<thead>
<tr>
<th>Function</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coalesce(value, value[, value ...])</td>
<td>Returns the first value not equal to NULL.</td>
</tr>
<tr>
<td>If(condition, value, value)</td>
<td>Evaluates the condition and returns the first value if it is True, and the second value otherwise.</td>
</tr>
<tr>
<td>Max(v1, v2[, v3 ...])</td>
<td>Computes the maximum value in the list of values. Accepts any number of values of any type. NULL values are ignored. If all values are NULL, returns NULL.</td>
</tr>
<tr>
<td>Min(v1, v2[, v3 ...])</td>
<td>Computes the minimum value in the list of values. Accepts any number of values of any type. NULL values are ignored. If all values are NULL, returns NULL.</td>
</tr>
<tr>
<td>NullIf(value, value)</td>
<td>Returns NULL if the first value is equal to the second value, and the first value otherwise.</td>
</tr>
<tr>
<td>TypeName(value)</td>
<td>Returns the name of a value type.</td>
</tr>
<tr>
<td>VarType(value)</td>
<td>Returns an integer value representing a value type.</td>
</tr>
</tbody>
</table>

Numeric Expressions and Functions

Numeric expressions are expressions that return numeric values. Numeric expressions can include numeric literals. Numeric expressions can include the following arithmetic operators (in order of increasing priority):

+ Addition
- Subtraction
/ Division
* Multiplication
Div Integer division, equivalent to VBScript \\ (*
Mod Modulo. Performs a division, retaining only the remainder. [Note: this is in line with VB.NET but different from VBScript, which first converts floating point numbers to integers via rounding and then performs the division.]
^ Exponentiation

Numeric expressions can be used with the following functions:

<table>
<thead>
<tr>
<th>Function</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abs(number)</td>
<td>Returns the absolute value of a number.</td>
</tr>
<tr>
<td>Acos(number)</td>
<td>Returns the arc cosine of a number.</td>
</tr>
</tbody>
</table>
Asin(number) Returns the arc sine of a number.
Atn(number) Returns the arc tangent of a number.
Atn2(y, x) Returns the arc tangent of y/x.
Ceil(x) Returns the smallest integer that is greater than or equal to x.
Cos(number) Returns the cosine of a number.
Deg2Rad(x) Converts x from degrees to radians.
Exp(number) Returns the exponent of a number.
Fix(number) Converts a fractional number to an integer number rounding towards zero.
Floor(x) Returns the largest integer that is less than or equal to x.
Int(number) Converts a fractional number to an integer number rounding down.
Log(number) Returns the natural logarithm of a number.
Log2(x) Returns the binary logarithm of x.
Log10(x) Returns the decimal logarithm of x.
Pow(x, y) Returns the value of x raised to the power of y.
Rad2Deg(x) Converts x from radians to degrees.
Rgb(red, green, blue) Composes red, green, and blue components of a color into an integer number.
Rnd Returns a random number between 0 and 1. Used without parentheses.
Round(number[, decimals]) Rounds a fractional number to a given number of decimal digits and returns the result. If the number of decimal digits is omitted, it is assumed to be 0.
Sgn(number) Returns the sign of a number.
Sin(number) Returns the sine of a number.
Sqr(number) Returns the square root of a number.
Tan(number) Returns the tangent of a number.

String Expressions

String expressions are expressions that return string values. String expressions can include string literals. String expressions can be concatenated with ||, +, or & operators. String expressions can be used with the following functions:

<table>
<thead>
<tr>
<th>Function</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit_Length(string)</td>
<td>Returns the length of a string in bits (an ANSI character is 8 bit wide and a Unicode character is 16 bit wide). Also supports binary values.</td>
</tr>
<tr>
<td>Char_Length(string)</td>
<td>Returns the length of a string in characters.</td>
</tr>
<tr>
<td>Character_Length(string)</td>
<td>Same as Char_Length(string).</td>
</tr>
<tr>
<td>Chr(number)</td>
<td>Returns a character with the given numeric value.</td>
</tr>
</tbody>
</table>
**FormatCurrency(number[, decimals[, leading[, parens[, groups]]]])**
Prints a currency value into a string using the given number of decimal digits (system-defined by default). The third parameter specifies whether a leading zero is included (-1) or excluded (0). The fourth parameter specifies whether negative values are printed in parentheses (-1) or not (0). The fifth parameter specifies whether the numbers are grouped (-1) or not (0). The third, fourth, and fifth parameters can be set to be taken from the regional settings of the current user (-2, the default).

**FormatDateTime(date[, format])**
Prints a date into a string using either the default format (0), long date format (1), short date format (2), long time format (3), or short time format (4, uses 24 hours).

**FormatNumber(number[, decimals[, leading[, parens[, groups]]]])**
Prints a number into a string using the given number of decimal digits (system-defined by default). The third parameter specifies whether a leading zero is included (-1) or excluded (0). The fourth parameter specifies whether negative values are printed in parentheses (-1) or not (0). The fifth parameter specifies whether the numbers are grouped (-1) or not (0). The third, fourth, and fifth parameters can be set to be taken from the regional settings of the current user (-2, the default).

**FormatPercent(number[, decimals[, leading[, parens[, groups]]]])**
Prints a percentage into a string using the given number of decimal digits (system-defined by default). The third parameter specifies whether a leading zero is included (-1) or excluded (0). The fourth parameter specifies whether negative values are printed in parentheses (-1) or not (0). The fifth parameter specifies whether the numbers are grouped (-1) or not (0). The third, fourth, and fifth parameters can be set to be taken from the regional settings of the current user (-2, the default).

**Hex(number)**
Prints an integer number into a string using hexadecimal notation.

**InStr(string, string[, start[, case]])**
Returns the position of the first occurrence of the second string in the first string, or 0 if there are no occurrences. The third parameter specifies the starting position of a search (1 by default). The fourth parameter specifies whether the search is case-sensitive (0, the default), or case-insensitive (1).

**InStrRev(string, string[, start[, case]])**
Returns the position of the last occurrence of the second string within the first string, or 0 if there are no occurrences. The third parameter specifies the starting position of a search (1 by default). The fourth parameter specifies whether the search is case-sensitive (0, the default), or case-insensitive (1).

**Left(string, length)**
Returns the given number of starting characters in a string.

**Len(string)**
Same as Char_Length(string).

**LCase(string)**
Converts all string characters to lower case and returns the result.

**Lower(string)**
Same as LCase(string).

**LTrim(string)**
Trims leading white space characters from a string and returns the result.

**Mid(string, start[, length])**
Returns the portion of a string starting with a given character (one-based). If the length parameter is omitted, the function returns the rest of the string.

**Oct(number)**
Prints an integer number into a string using octal notation.

**Octet_Length(string)**
Returns the length of a string in bytes (an ANSI character is 1 byte wide and a Unicode character is 2 byte wide). Also supports binary values.

**RegExp(string, pattern, substitute)**
Transforms a string with regular expressions and returns the result.

**Replace(string, what, with[, start[, count[, case]]])**
Replaces all occurrences of one substring in a string with another substring, and returns the result. The fourth parameter specifies the starting position of a search (1 by default). The
fifth parameter specifies the maximum number of substitutions to perform and can be set to -1 (the default) to perform all possible substitutions. The sixth parameter specifies whether the search is case-sensitive (0, the default), or case-insensitive (1).

**Right(string, length)** Returns the given number of trailing characters in a string.

**RTrim(string)** Trims trailing white space characters from a string and returns the result.

**SoundsLike(string, string)** Returns True if strings sound like each other, and False otherwise. Uses Soundex algorithm for English language.

**Space(number)** Returns a string with the given number of spaces.

**StrComp(string, string[, case])** Compares two strings and returns -1 if the first string is less than the second string, 1 if the first string is greater than the second string, and 0 if the strings are equal. The third parameter specifies whether the comparison is case-sensitive (0, the default), or case-insensitive (1).

**String(number, character)** Returns a string with the given number of given characters. The second parameter can be either a string (that gets reduced to the first character), or a number (that gets converted to a string with a Chr function).

**StrReverse(string)** Reverses a string and returns the result.

**Token(string[, index[, delimiters]]** Returns a token with the given index (one-based). The third parameter is a string that contains token delimiters. If the third parameter is omitted, tokens are separated with white space characters.

**UCase(string)** Converts all string characters to upper case and returns the result.

**Upper(string)** Same as UCase(string).

**Date Expressions**

Date expressions are expressions that return date values. Date expressions can include date literals. Date expressions can be used with the following functions:

<table>
<thead>
<tr>
<th>Function</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Current_Date</strong></td>
<td>Same as Date. Used without parentheses.</td>
</tr>
<tr>
<td><strong>Current_Time</strong></td>
<td>Same as Time. Used without parentheses.</td>
</tr>
<tr>
<td><strong>Current_TimeStamp</strong></td>
<td>Same as Time. Used without parentheses.</td>
</tr>
<tr>
<td><strong>Date</strong></td>
<td>Returns the current date. Used without parentheses.</td>
</tr>
<tr>
<td><strong>DateAdd(interval, number, date)</strong></td>
<td>Increases or decreases a date by the given value in the given time interval, and returns the result. The first parameter is a string that specifies which interval to increase (&quot;yyyy&quot; for year, &quot;q&quot; for quarter, &quot;m&quot; for month, &quot;y&quot; for day of year, &quot;d&quot; for day of month, &quot;w&quot; for weekday, &quot;ww&quot; for week of year, &quot;h&quot; for hour, &quot;n&quot; for minute, and &quot;s&quot; for second).</td>
</tr>
<tr>
<td><strong>DateDiff(interval, date, date[, start[, week]]</strong></td>
<td>Returns the number of time intervals between two dates. The first parameter is a string that specifies the time interval. The fourth parameter is the numeric index of a day that starts the week (0 for the default start, 1 for Sunday, 2 for Tuesday, and so on). The fifth parameter is a number that specifies the method used to determine the first week in a year (0 for the default method, 1 to use the week that contains the 1st of January, 2 to use the first week that contains at least 4 days in the new year, 3 to use the first week that is entirely in the new year).</td>
</tr>
</tbody>
</table>
DatePart(interval, date[, start[, week]]) Returns the value of a specified time interval in a date. The first parameter is a string that specifies the time interval. The third parameter is the numeric index of a day that starts the week. The fourth parameter is a number that specifies the method used to determine the first week in a year.

DateSerial(year, month, day) Composes a date without a time.

DateValue(value) Converts a value to a date without a time ignoring hour, minute, and second information.

Day(date) Returns the day of month of a date (1 to 31).

Hour(date) Returns the hour of a date (0 to 23).

Minute(date) Returns the minute of a date (0 to 59).

Month(date) Returns the month of a date (1 to 12).

MonthName(month[, abbreviate]) Returns the name of a month (1 to 12). The second parameter switches between abbreviated names (True) and full names (False, the default).

Now Same as Time. Used without parentheses.

Second(date) Returns the second of a date (0 to 59).

Time Returns the current date and time. Used without parentheses.

Timer Returns the number of seconds since 12:00 AM. Used without parentheses.

TimeSerial(hour, minute, second) Composes the time portion of a date.

TimeValue(value) Converts a value to the time portion of a date ignoring year, month, and day information.

Weekday(date[, start]) Returns the weekday of a date (1 to 7). The second parameter is the numeric index of a day that starts the week (0 for the default start, 1 for Sunday, 2 for Tuesday, and so on).

WeekdayName(day[, abbreviation[, start]]) Returns the name of a weekday (1 to 7). The second parameter switches between abbreviated names (True) and full names (False, the default). The third parameter is the numeric index of a day that starts the week.

Year(date) Returns the year of a date.

Boolean Expressions

Boolean expressions are expressions that return Boolean values. Boolean expressions can include Boolean literals, such as True and False, or Yes and No. Boolean expressions can include the following comparison operators:

> Greater than
< Less than
<> Not equal to
>= Greater than or equal to
<= Less than or equal to
= Equal to
When applied to strings, the above operators work from the leftmost to the rightmost character, comparing characters in alphabetical order (typically, A is less than B, B is less than C, digits are less than letters, upper case letters are less than lower case letters, etc.)

Boolean expressions can also include the following Boolean operators:

- **And**: Logical AND (True if both operands are True)
- **Or**: Logical OR (True if either operand is True)
- **Not**: Logical NOT (True if the operand is False)
- **Eqv**: Logical equivalence (True if both operands are either True or False)
- **Imp**: Logical implication (True unless the first operand is True and the second operand is False)
- **Xor**: Logical XOR (True if one of the operands is True, and another is False)

**Conversion Functions**

The expressions can also include the following functions that convert values from one type to another:

<table>
<thead>
<tr>
<th>Function</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>BinaryToHex(value)</td>
<td>Converts a typed (for example, Geom) or untyped binary value to a text value, printing each byte using hexadecimal notation.</td>
</tr>
<tr>
<td>CBool(value)</td>
<td>Converts a value to a Boolean.</td>
</tr>
<tr>
<td>CByte(value)</td>
<td>Converts a value to a 8 bit unsigned integer number (byte).</td>
</tr>
<tr>
<td>CCoordSys(value)</td>
<td>Converts a value to a coordinate system.</td>
</tr>
<tr>
<td>CCur(value)</td>
<td>Converts a value to a currency.</td>
</tr>
<tr>
<td>CDate(value)</td>
<td>Converts a value to a date.</td>
</tr>
<tr>
<td>CDbl(value)</td>
<td>Converts a value to a double precision floating-point number.</td>
</tr>
<tr>
<td>CGeom (value)</td>
<td>Converts a value to a geometric object of type Geometry.</td>
</tr>
<tr>
<td>CGeomSDE(value)</td>
<td>Converts given data to geometric object of type Geometry (SDE).</td>
</tr>
<tr>
<td>CGeomSHP(value)</td>
<td>Converts given data to geometric object of type Geometry (SHP).</td>
</tr>
<tr>
<td>CGeomWKB(value)</td>
<td>Converts given data to geometric object of type Geometry (WKB).</td>
</tr>
<tr>
<td>CInt(value)</td>
<td>Converts a value to a 32 bit integer number.</td>
</tr>
<tr>
<td>CLng(value)</td>
<td>Converts a value to a 32 bit integer number (same as CInt).</td>
</tr>
<tr>
<td>CSng(value)</td>
<td>Converts a value to a single precision floating-point number.</td>
</tr>
<tr>
<td>CStr(value)</td>
<td>Converts a value to a string.</td>
</tr>
<tr>
<td>HexToBinary(value)</td>
<td>Converts a text value in the hexadecimal notation format used by the BinaryToHex function to an untyped binary value.</td>
</tr>
</tbody>
</table>

The **BinaryToHex** and **HexToBinary** functions are used to convert binary data to text format and vice versa. They are especially useful for storing binary data within data sources such as DBF files that do not support binary data columns.
Lookup Values and CStr and CAST

Invoking CStr on a lookup value or using CAST to convert the value to a string returns the descriptive name of the lookup value.

Example

We have a table T with a lookup column Region with values East and West. We want to select all records with the value of Region being East. We can do this with the following query:

```
SELECT * FROM [T] WHERE CStr([Region]) = "East";
```

Example

We have a drawing D and we want to select all areas it contains. We can do this with the following query:

```
SELECT * FROM [D] WHERE CAST([Type (I)]) AS TEXT) = "Area";
```

See Also

Data Types
Regular Expressions
Spatial Extensions
Geocoding Extensions
Raster Extensions

Historical Note

The term "Boolean" takes its name from George Boole (1815 -1864), one of the great mathematicians of the nineteenth century and the inventor of mathematical logic and Boolean algebra, a systematic means of writing and manipulating logical concepts using mathematical symbols. Boolean algebra is the foundation of digital computer architecture.

Boole wrote fifty papers, two textbooks and two major volumes of work on mathematical logic and was a professor at Queen's College in Cork despite never having attended college or earning a degree. He was entirely self-taught. The son of a poor London shoemaker, Boole was encouraged by his father to pursue an early interest in learning. By age 12 he had mastered Latin and Greek and later turned his attention to mathematics. He died at 49 from an illness apparently incurred as a result of giving a lecture in soaking wet clothes after walking two miles through the rain. His untimely death interrupted a very productive career. One wonders what marvels the world lost for lack of an umbrella.

In English, George Boole's name is pronounced with a silent "e" to rhyme with "tool" or "cool".
Aggregate Functions

An aggregate function is a function that calculates the total value of a group of values. Aggregate functions are commonly used with the GROUP BY clause, and with the TRANSFORM statement.

The following table lists available aggregate functions:

<table>
<thead>
<tr>
<th>Function</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>AllBranches</td>
<td>Combines all branches of geometric objects into a single object. See the Spatial Extensions topic.</td>
</tr>
<tr>
<td>AllCoords</td>
<td>Combines all coordinates of geometric objects into a single object. See the Spatial Extensions topic.</td>
</tr>
<tr>
<td>Avg</td>
<td>Calculates the arithmetic mean of a set of numeric values.</td>
</tr>
<tr>
<td>Count</td>
<td>Calculates the number of values in a set of values of any type.</td>
</tr>
<tr>
<td>First</td>
<td>Retrieves the first value in a set of values of any type.</td>
</tr>
<tr>
<td>Last</td>
<td>Retrieves the last value in a set of values of any type.</td>
</tr>
<tr>
<td>Max</td>
<td>Calculates the maximum value in a set of numeric, string, or date values.</td>
</tr>
<tr>
<td>Min</td>
<td>Calculates the minimum value in a set of numeric, string, or date values.</td>
</tr>
<tr>
<td>StDev</td>
<td>Calculates the standard deviation of a set of numeric values. Requires at least two values.</td>
</tr>
<tr>
<td>StDevP</td>
<td>Calculates the unadjusted standard deviation of a set of numeric values.</td>
</tr>
<tr>
<td>Sum</td>
<td>Calculates the sum of a set of numeric values.</td>
</tr>
<tr>
<td>UnionAll</td>
<td>Unions all areas into a single area. See the Spatial Extensions topic.</td>
</tr>
<tr>
<td>Var</td>
<td>Calculates the variation in a set of numeric values. Requires at least two values.</td>
</tr>
<tr>
<td>VarP</td>
<td>Calculates the unadjusted variation in a set of numeric values.</td>
</tr>
</tbody>
</table>

All aggregate functions ignore NULL values. If the set of values is empty, the Count function returns 0, and all other aggregate functions return NULL.

Examples

This example uses the Avg aggregate function to calculate the average freight charges for orders with freight charges over $100.

```
SELECT Avg([Freight]) AS [Average Freight] FROM [Orders] WHERE [Freight] > 100;
```

This example uses the Sum, Avg, and StDev aggregate functions to calculate the total number of units, the average number of units, and the standard deviation of the number of units ordered for each product:

```
SELECT [Product Name], Sum([Quantity]) AS [Total Quantity], Avg([Quantity]) AS [Typical Quantity], StDevP([Quantity]) AS [Standard Deviation] FROM [Products] INNER JOIN [Order Details] ON [Order Details].[Product ID] = [Products].[Product ID] GROUP BY [Product Name];
```
This example uses the Max and Min aggregate functions to calculate the maximum and minimum order adjusted for discount, and the maximum discount made on each product:

```sql
SELECT [Product Name], Min([Order Details].[Unit Price] * (1-[Discount]) * [Quantity]) AS [Min Value], Max([Order Details].[Unit Price] * (1-[Discount]) * [Quantity]) AS [Max Value], Max([Discount]) AS [Max Discount] FROM [Products] INNER JOIN [Order Details] ON [Order Details].[Product ID] = [Products].[Product ID] GROUP BY [Product Name];
```

This example uses the Max and Min aggregate functions to select the first and the last order for each product in 1992:

```sql
SELECT [Product Name], Min([Order Date]) AS [First Order], Max([Order Date]) AS [Last Order] FROM ([Orders] INNER JOIN [Order Details] ON [Order Details].[Order ID] = [Orders].[Order ID]) INNER JOIN [Products] ON [Products].[Product ID] = [Order Details].[Product ID] WHERE Year([Order Date]) = 1992 GROUP BY [Product Name];
```

**Statements**

**ALTER TABLE Statement**

Modifies the design of an existing table.

**Syntax**

```sql
ALTER TABLE table {
    ADD [COLUMN] column type [(size)] [DEFAULT default] |
    ALTER [COLUMN] column type [(size)] [DEFAULT default] |
    ALTER [COLUMN] column SET DEFAULT default |
    DROP [COLUMN] column |
    RENAME [COLUMN] column TO columnNew
}
```

The ALTER TABLE statement has these parts:

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>table</td>
<td>The name of the table to be altered.</td>
</tr>
<tr>
<td>column</td>
<td>The name of the column to be altered or added to or deleted from table.</td>
</tr>
<tr>
<td>columnNew</td>
<td>The new name of the altered column.</td>
</tr>
<tr>
<td>type</td>
<td>The data type of column.</td>
</tr>
<tr>
<td>size</td>
<td>The size of the altered column in characters or bytes for text or binary columns.</td>
</tr>
<tr>
<td>default</td>
<td>An expression defining the new default value of the altered column. Can contain literal values, and functions of these values.</td>
</tr>
</tbody>
</table>

**Remarks**

Using the ALTER TABLE statement, we can alter an existing table in several ways. We can:

- Use ADD COLUMN to add a new column to the table. Specify the name, data type, an optional size, and an optional default value of the column.
- Use ALTER COLUMN to alter type, size or default value of an existing column.
- Use DROP COLUMN to delete a column. Specify only the name of the column.
- Use RENAME COLUMN to rename an existing column.

We cannot add, delete or modify more than one column at a time.

The ALTER TABLE statement is an action query and cannot be used from within the Manifold ODBC driver.

**Examples**

Add a currency Salary column to the Employees table:

```
ALTER TABLE [Employees] ADD COLUMN [Salary] CURRENCY;
```

Remove the Salary column from the Employees table. (Oh no!...):

```
ALTER TABLE [Employees] DROP COLUMN [Salary];
```

Add a text Rank column with the default value of "High" to the Employees table:

```
ALTER TABLE [Employees] ADD COLUMN [Rank] TEXT DEFAULT "High";
```

Change the default value of the Rank column added above to "Medium":

```
ALTER TABLE [Employees] ALTER COLUMN [Rank] SET DEFAULT "Medium";
```

Change the type of the Rank column added above to fixed-length text with 10 characters:

```
ALTER TABLE [Employees] ALTER COLUMN [Rank] TEXT(10);
```
CREATE DRAWING Statement

Creates a new table.

Syntax

CREATE DRAWING drawing [(column type [(size)] [DEFAULT default] [, column type [(size)] ...])]

The CREATE DRAWING statement has these parts:

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>drawing</td>
<td>The name of the drawing to be created.</td>
</tr>
<tr>
<td>column</td>
<td>The name of the column or columns to be created in the table bound to the new drawing, in addition to the ID column and intrinsic columns.</td>
</tr>
<tr>
<td>type</td>
<td>The data type of the column in the table bound to the new drawing.</td>
</tr>
<tr>
<td>size</td>
<td>The column size in characters or bytes for text or binary columns.</td>
</tr>
<tr>
<td>default</td>
<td>The expression defining the default value of column in the table bound to the new drawing. Can contain literal values, and functions of these values.</td>
</tr>
</tbody>
</table>

Remarks

Use the CREATE DRAWING statement to define a new drawing. The CREATE DRAWING statement is similar to the CREATE TABLE statement, except that it creates a drawing together with the ID column and the intrinsic columns, and does not require supplying extra columns. All text columns are created as Unicode text type columns.

The CREATE DRAWING statement is an action query and cannot be used from within the ODBC driver.

To define a new table not bound to a drawing, use the CREATE TABLE statement.

Examples

This example creates a new drawing called "This Drawing" with only the ID column and intrinsic columns:

```
CREATE DRAWING [This Drawing];
```

This example creates a new drawing called "This Drawing" with the ID column, intrinsic columns and an integer column:

```
CREATE DRAWING [This Drawing] ([Population] INTEGER);
```
CREATE TABLE Statement
Creates a new table.

Syntax

```
CREATE TABLE table (column type [(size)] [DEFAULT default]
[, column type [(size)] ...])
```

The CREATE TABLE statement has these parts:

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>table</td>
<td>The name of the table to be created.</td>
</tr>
<tr>
<td>column</td>
<td>The name of the column or columns to be created in the new table. We must create at least one column.</td>
</tr>
<tr>
<td>type</td>
<td>The data type of the column in the new table.</td>
</tr>
<tr>
<td>size</td>
<td>The column size in characters or bytes for text or binary columns.</td>
</tr>
<tr>
<td>default</td>
<td>The expression defining the default value of column in the new table. Can contain literal values, and functions of these values.</td>
</tr>
</tbody>
</table>

Remarks

Use the CREATE TABLE statement to define a new table and its columns. All text columns are created as Unicode text type columns.

The CREATE TABLE statement is an action query and cannot be used from within the ODBC driver.

To define a new drawing, use the CREATE DRAWING statement.

Examples

This example creates a new table called "This Table" with two text columns.

```
CREATE TABLE [This Table] ([First Name] TEXT, [Last Name] TEXT);
```

This example creates a new table called "My Table" with two text columns and a date/time column.

```
CREATE TABLE [My Table] ([First Name] TEXT, [Last Name] TEXT, [Birth Date] DATETIME);
```

This example creates a new table with two text columns of specified length (20 and 40 characters respectively) and an integer column with a default value of -1.

```
CREATE TABLE [New Table] ([First Name] TEXT(20), [Last Name] TEXT(40), [SSN] INTEGER DEFAULT -1);
```

This example creates a new table called "Orders" with a text column, an integer column with a default value of 1, and a date/time column with a default value equal to the 20th of September 2003:

```
CREATE TABLE [Orders] ([Product] TEXT, [Quantity] INTEGER DEFAULT 1, [Date] DATETIME DEFAULT #20/09/2003#);
```
CREATE VIEW Statement
Creates a new query.

Syntax

CREATE VIEW query AS text

The CREATE VIEW statement has these parts:

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>query</td>
<td>The name of the query to be created.</td>
</tr>
<tr>
<td>text</td>
<td>The text of the new query. Should not be an action query</td>
</tr>
</tbody>
</table>

Remarks

Use the CREATE VIEW statement to create a new query that returns records.

The CREATE VIEW statement is an action query and can not be used from within the ODBC driver.

The CREATE VIEW query statement allows creating a query referring to a table or column that does not yet exist. Obviously, this capability should be used with reasonable care and foresight.

Examples

This example creates a new query named "Large Cities" that selects cities whose population exceeds 1,000,000 from the "Cities" table:

```
CREATE VIEW [Large Cities] AS
SELECT * FROM [Cities] WHERE [Population] > 1000000;
```

This example creates a new query named "Large American Cities" that selects American cities from the "Large Cities" query created above:

```
CREATE VIEW [Large American Cities] AS
SELECT * FROM [Large Cities] WHERE [Region] = "North America";
```
DELETE Statement
Removes records from a specified table.

Syntax

```sql
DELETE FROM table [WHERE ...]
```

The DELETE statement has these parts:

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>table</td>
<td>The name of the table from which records are deleted.</td>
</tr>
</tbody>
</table>

Remarks

The DELETE statement is especially useful when we want to delete many records. Unlike the UPDATE statement that can operate on queries, the DELETE statement can only delete records from tables.

To drop an entire table use the DROP TABLE statement. If we delete the table, however, the structure is lost. In contrast, when we use DELETE, only the data is deleted; the table structure and all of the table properties, such as column formatting and transfer rules, remain intact.

The DELETE statement deletes entire records, not just data in specific columns. If we want to delete values in a specific column, we use the UPDATE statement.

The DELETE statement is an action query and cannot be used from within the Manifold ODBC driver.

Important

After we remove records using a DELETE query, we can't undo the operation. If we want to know which records were deleted, first examine the results of a SELECT query that uses the same criteria, and then run the DELETE query.

Maintain backup copies of data at all times. If we delete the wrong records, we can retrieve them from our backup copies.

Example

Delete all records for employees whose title is Trainee:

```sql
DELETE FROM [Employees] WHERE [Title] = "Trainee";
```
DROP DRAWING Statement

Deletes an existing table.

Syntax

DROP DRAWING drawing

The DROP DRAWING statement has these parts:

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>drawing</td>
<td>The name of the drawing to be deleted, or the name of the table bound to the drawing to be deleted, or the name of a child theme of the drawing to be deleted.</td>
</tr>
</tbody>
</table>

Remarks

We can use CREATE DRAWING to create both a drawing and a table bound to that drawing and ALTER TABLE to modify the table.

We cannot use DROP DRAWING to delete a query. To delete a query, use DROP VIEW.

The DROP DRAWING statement is an action query and cannot be used from within the Manifold ODBC driver.

Example

Delete a drawing named "Cities":

DROP DRAWING [Cities];
**DROP TABLE Statement**
Deletes an existing table.

**Syntax**

```
DROP TABLE table
```

The DROP TABLE statement has these parts:

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>table</td>
<td>The name of the table to be deleted.</td>
</tr>
</tbody>
</table>

**Remarks**

We can use CREATE TABLE to create a table and ALTER TABLE to modify a table.

We cannot use DROP TABLE to delete a query. To delete a query, use DROP VIEW.

The DROP TABLE statement is an action query and cannot be used from within the Manifold ODBC driver.

Attempting to delete a table bound to a drawing using DROP TABLE will fail with a descriptive error message.

**Example**

Delete a table named "Employees":

```
DROP TABLE [Employees];
```
DROP VIEW Statement
Deletes an existing query.

Syntax

DROP VIEW query

The DROP VIEW statement has these parts:

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>query</td>
<td>The name of the query to be deleted.</td>
</tr>
</tbody>
</table>

Remarks

We can use CREATE VIEW to create a query.

We cannot use DROP VIEW to delete a table. To delete a table, use DROP TABLE.

The DROP VIEW statement is an action query and cannot be used from within the Manifold ODBC driver.

Example

Delete a query named "Youngest Employee":

```sql
DROP VIEW [Youngest Employee];
```
INSERT INTO Statement

Adds a record or multiple records to a table.

Syntax

```
INSERT INTO table {DEFAULT VALUES | [(column [, column ...])]} query
```

The INSERT INTO statement has these parts:

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>table</td>
<td>The name of the table to which records are to be appended.</td>
</tr>
<tr>
<td>column</td>
<td>The name of a column in the target table.</td>
</tr>
<tr>
<td>query</td>
<td>A SELECT query.</td>
</tr>
</tbody>
</table>

Remarks

We can use the INSERT INTO statement to insert either a single record with default values, or to insert one or more records with arbitrary values. Records are added to the end of the table.

The list of columns specifies the columns to insert to. Columns in the target table that are not mentioned in the list of columns are supplied with default values. Omitting the list of columns is equivalent to specifying all columns in the order they appear in the target table.

Using the INSERT INTO statement with a SELECT query that returns a table inserts the records returned by the query. If the number of columns in the returned table does not match the number of columns in the list of target columns, the statement will fail. The columns in the returned table are matched to those in the list of target columns using their order, that is, the first column in the returned table is matched to the first target column, the second column in the returned table is matched to the second target column and so on. The types and sizes of columns in the returned table must be compatible with those of the target columns. The names of columns are ignored.

Using the INSERT INTO statement with a VALUES variation of a SELECT query inserts the literal values specified in the query. If the number of supplied values is not an even multiple of the number of columns in a column list, or not an even multiple of the number of columns in the target table if the column list is omitted, the statement will fail.

To create a new table, use the SELECT INTO statement.

To find out which records will be appended before we run the INSERT INTO query, we first execute and view the results of a SELECT query that uses the same selection criteria.

The INSERT INTO statement is an action query and cannot be used from within the Manifold ODBC driver.

Examples

Create a new blank record in the "Customers" table:

```
INSERT INTO [Customers] DEFAULT VALUES;
```

Select all records in a hypothetical "New Customers" table and add them to the "Customers" table. When individual columns are not designated, the number of columns in the table returned by the SELECT must match that in the target table:

```
INSERT INTO [Customers] SELECT * FROM [New Customers];
```
This example is identical to the above but uses simplified syntax for the SELECT portion:

```
INSERT INTO [Customers] TABLE [New Customers];
```

Create a new record in the "Employees" table. The first name is Harry, the last name is Washington, and the job title is Trainee:

```
INSERT INTO [Employees] ([First Name], [Last Name], [Title]) VALUES ("Harry", "Washington", "Trainee");
```

Add a point at the 0,0 coordinate location to an existing drawing called "Drawing" using the \texttt{Geom (I)} intrinsic field:

```
INSERT INTO [Drawing] ([Geom (I)])
VALUES (CGeom(CGeomWKB("point (0 0)")));
**SELECT Statement**

Selects data from existing tables or queries.

**Syntax**

\[
\text{atom \{UNION | EXCEPT | INTERSECT\} [ALL] atom ...} \] [ORDER BY ...]
\]

where \text{atom} is:

\[
\text{SELECT [ALL | DISTINCT | DISTINCTROW] [SKIP n [PERCENT]]} \]
\[
\text{[TOP n [PERCENT]]} \] \text{column [, column ...]} \]
\[
\text{FROM table [, table ...]} \]
\[
\text{[WHERE ...]} \] [SPLIT BY ...] [LEAVING ...] [GROUP BY ...] \]
\[
\text{[HAVING ...]} \]
\]
or:

\[
\text{TABLE table} \]
\]
or:

\[
\text{VALUES \{value | NULL\[, \{value | NULL\} ...\]} [, \{(value | NULL\[, \{value | NULL\} ...\]} ...\]} [NAMES (name[, name ...])] \]
\]

The SELECT statement has these parts:

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>column</td>
<td>The name of a column or an asterisk, possibly qualified with the name of a table as in [Table].[Column], or [Table]*, or an expression. Can include an alias. A non-qualified asterisk selects all columns, and must be the only item in the column list.</td>
</tr>
<tr>
<td>n</td>
<td>The number of records in SKIP or TOP or percent in SKIP PERCENT or TOP PERCENT.</td>
</tr>
<tr>
<td>name</td>
<td>The name to use for a column.</td>
</tr>
<tr>
<td>table</td>
<td>The name of a table containing the data for a TABLE variant. The name of a table containing data (with a possible alias, and column filter), or another query, or several tables or queries joined together with one of the join operators, for a SELECT variant.</td>
</tr>
<tr>
<td>value</td>
<td>A literal value, or an expression, or a query that returns a single value.</td>
</tr>
</tbody>
</table>

**Remarks**

SELECT statements don't change data. The minimum syntax for a SELECT statement is:

\[
\text{SELECT columns FROM table;}
\]

We can use an asterisk (*) to select all columns in a table. The following example selects all columns in the Employees table:

\[
\text{SELECT * FROM [Employees];}
\]
We may restrict the table to return only the columns we want with a table filter. The following example restricts the output of the above query to just the "First Name," and "Last Name" columns:

```sql
SELECT [First Name], [Last Name] FROM [Employees];
```

Alternatively, we may specify the columns we want in the FROM clause. The following example is identical to the above query but uses the FROM clause:

```sql
SELECT * FROM [Employees] ([First Name], [Last Name]);
```

If a column name is included in more than one table in the FROM clause, precede it with the table name and a dot (.). In the following example, the "Category ID" column is in both the "Products" table and the "Categories" table. The statement selects product names from the "Products" table and category names from the "Categories" table:

```sql
SELECT [Products].[Product Name], [Categories].[Category Name]
FROM [Products] INNER JOIN [Categories]
ON [Products].[Category ID] = [Categories].[Category ID];
```

When a data set is returned in answer to a query, the SQL engine uses column names from the originating tables in the data set returned. If we want a different column name or a name isn’t implied by the expression used to generate the column, we can use column aliases. The following example renames the returned column "Birth Date" to just "Birth":

```sql
SELECT [Birth Date] AS [Birth] FROM [Employees];
```

The AS keyword is optional so we might rewrite the above query as:

```sql
SELECT [Birth Date] [Birth] FROM [Employees];
```

However, because of the clarity the AS keyword brings to a query it is madness not to use it. Keep in mind that queries we write today might not be so obvious in intent when we reread them in the future.

In addition to providing alternate names for columns, we could also provide alternate names for tables. This can be useful for shortening expressions inside the query and for handling complex joins. The following example displays the name of the category for each product shortening the names of the involved tables for clarity:

```sql
SELECT [P].[Product Name], [C].[Category Name]
ON [P].[Category ID] = [C].[Category ID];
```

Whenever we write queries that return ambiguous or duplicate column or table names, we must use aliases to provide alternate names.

The TABLE variant of the SELECT statement returns all data in a given table. The following statements are equivalent:

```sql
SELECT * FROM [Employees];
TABLE [Employees];
```

The VALUES variant of the SELECT statement synthesizes a table from the supplied values and is typically used with INSERT INTO or with sub-queries. The following statement synthesizes a table with a single row and two columns with values 1 and "ABC" (note the use of the concatenation operator in the second value):

```sql
VALUES (1, "AB" & "C");
```
The WHERE, UNION, and other components of the SELECT statement restrict and organize the returned data. For more information, see the Help topic for the component being using.

**Examples**

Select the "Product Name," and "Unit Price" columns of all records in the "Products" table:

```sql
SELECT [Product Name], [Unit Price] FROM [Products];
```

Count the total number of products and name the returned column "Total Products":

```sql
SELECT Count(*) AS [Total Products] FROM [Products];
```

Show the minimum, maximum, and average prices for products:

```sql
SELECT Min([Unit Price]) AS [Min Price], Max([Unit Price]) AS [Max Price],
Avg([Unit Price]) AS [Avg Price] FROM [Products];
```

Create a table with one record and three integer columns with values 1, 2, 3:

```sql
VALUES (1, 2, 3)
```

Create a table with one integer column and three records with values 1, 2, 3:

```sql
VALUES (1), (2), (3)
```

Create a table with three records, an integer column and a text column:

```sql
VALUES (1, "a"), (2, "b"), (3, "c")
```

Create a table with two integer columns named "x" and "y" and three records:

```sql
VALUES (1, 1), (2, 2), (3, 3) NAMES (x, y)
```

**See Also**

ALL, DISTINCT, SKIP, TOP Quantifiers
**SELECT INTO Statement**

Selects data from existing tables or queries into a new table.

**Syntax**

```
SELECT [ALL | DISTINCT | DISTINCTROW | TOP n [PERCENT]]
column [, column ...]
INTO tableNew FROM table [, table ...]
[WHERE ...] [SPLIT BY ...] [LEAVING ...] [GROUP BY ...]
[HAVING ...] [ORDER BY...]
```

The SELECT INTO statement has these parts:

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>n</td>
<td>The number of records or percent in TOP or TOP PERCENT</td>
</tr>
<tr>
<td>column</td>
<td>The name of a column or an asterisk, possibly qualified with the name of a table as in [Table].[Column], or [Table]*, or an expression. Can include an alias. A non-qualified asterisk selects all columns, and must be the only item in the column list.</td>
</tr>
<tr>
<td>table</td>
<td>The name of the table containing the data (with a possible alias, and column filter), or another query, or several tables or queries joined together with one of the join operators.</td>
</tr>
<tr>
<td>tableNew</td>
<td>The name of the table to be created (must conform to standard naming conventions). If the name of the new table is the same as the name of an existing table, the statement fails.</td>
</tr>
</tbody>
</table>

**Remarks**

We can use SELECT INTO queries to archive records, make backup copies of our tables, or make snapshots of complex or frequently-changing data for reports. For example, we could produce a Monthly Sales by Region report by running the same SELECT INTO query each month.

When we use SELECT INTO to create a new table, the columns in the new table inherit the data type and size, the formatting, and the transfer rules of each column in the original tables.

To add data to an existing table, use the INSERT INTO statement.

To find out which records will be selected before we run the SELECT INTO query, we can first examine the results of a SELECT statement that uses the same selection criteria.

The SELECT INTO statement is an action query and cannot be used from within the Manifold ODBC driver.

**Example**

Select all records in the "Employees" table and copies them into a new table named "Backup Copy":

```
SELECT * INTO [Backup Copy] FROM [Employees];
```
TRANSFORM Statement
Selects data from one or more tables or queries grouping it with a specified pivot expression.

Syntax
TRANSFORM aggregate query PIVOT expression [IN list];

The TRANSFORM statement has these parts:

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>aggregate</td>
<td>An aggregate function that operates on the selected data.</td>
</tr>
<tr>
<td>query</td>
<td>A SELECT query with a GROUP BY clause.</td>
</tr>
<tr>
<td>expression</td>
<td>An expression used to group records.</td>
</tr>
<tr>
<td>list</td>
<td>A list of values allowed for the grouping expression.</td>
</tr>
</tbody>
</table>

Remarks
Use the TRANSFORM statement to group records by a pivot expression and to compute an aggregate function in each group.

Use the TRANSFORM statement with a query that includes the GROUP BY clause to establish two independent sets of groups and to compute an aggregate function in the records defined by the intersection of these sets of groups. The TRANSFORM statement splits each group returned by the GROUP BY clause into subgroups using a pivot expression, computes an aggregate function in each subgroup, and composes the computed values into a record for this group.

We can restrict the number of groups created by the pivot expression by supplying a list of possible values for this expression. Omitting a list of possible values will create a group for each different value of the expression.

Examples
This example uses the TRANSFORM statement to show the number of products in each category for each supplier:

TRANSFORM Count(*) SELECT [Supplier ID] FROM [Products] GROUP BY [Supplier ID] PIVOT [Category ID];

This example is similar to the example above but displays the name of each category and each supplier instead of their IDs:

TRANSFORM Count(*) SELECT [Company Name] FROM ([Products] INNER JOIN [Categories] ON [Categories].[Category ID] = [Products].[Category ID]) INNER JOIN [Suppliers] ON [Suppliers].[Supplier ID] = [Products].[Supplier ID] GROUP BY [Company Name] PIVOT [Category Name];

This example is similar to the example above but restricts product categories to beverages and seafood:

TRANSFORM Count(*) SELECT [Company Name] FROM ([Products] INNER JOIN [Categories] ON [Categories].[Category ID] = [Products].[Category ID]) INNER JOIN [Suppliers] ON [Suppliers].[Supplier ID] = [Products].[Supplier ID] GROUP BY [Company Name] PIVOT [Category Name] IN ("Beverages", "Seafood");

This example uses the TRANSFORM statement to show the number of orders taken by each employee for each calendar quarter of 1992:
TRANSFORM Count(*) SELECT [First Name] & " " & [Last Name] AS [Name] FROM [Employees] INNER JOIN [Orders] ON [Orders].[Employee ID] = [Employees].[Employee ID] WHERE Year([Order Date]) = 1992 GROUP BY [First Name] & " " & [Last Name] PIVOT "Q" & DatePart("q", [Order Date]);

This example uses the TRANSFORM statement to show the total dollar amount of orders taken by each employee for each calendar quarter of 1992:

TRANSFORM Sum([Unit Price] * [Quantity]) SELECT [First Name] & " " & [Last Name] AS [Name] FROM [Employees] INNER JOIN ([Orders] INNER JOIN [Order Details] ON [Order Details].[Order ID] = [Orders].[Order ID]) ON [Orders].[Employee ID] = [Employees].[Employee ID] WHERE Year([Order Date]) = 1992 GROUP BY [First Name] & " " & [Last Name] PIVOT "Q" & DatePart("q", [Order Date]);
**UPDATE Statement**

Changes values in columns in a specified table or query.

**Syntax**

```
UPDATE table SET column = {expression | DEFAULT} [, column = {expression | DEFAULT} ...] [WHERE ...]
```

The UPDATE statement has these parts:

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>table</td>
<td>The name of the table or query containing the data we want to modify.</td>
</tr>
<tr>
<td>column</td>
<td>The name of the column we want to modify.</td>
</tr>
<tr>
<td>expression</td>
<td>An expression with which to replace the old value. Can be a query that returns a single value.</td>
</tr>
</tbody>
</table>

**Remarks**

The UPDATE statement is especially useful when we want to change many records or when the records that we want to change are in multiple tables. If the records we want to change are in multiple tables, we create a SELECT query joining these tables together with UNION, and then use the UPDATE query against the SELECT.

Use the DEFAULT keyword to set the column to its default value.

If the result of the expression on the right side of the assignment operator is NULL, the value in the respective column is left unchanged.

We can change several columns at the same time. The following example increases the unit price by 10 percent and the reorder level by 50 for products with low number of units in stock:

```
UPDATE [Products] SET [Unit Price] = [Unit Price] * 1.1,
[Reorder Level] = [Reorder Level] + 50
WHERE [Units In Stock] <= 3;
```

The UPDATE statement is an action query and cannot be used from within the Manifold ODBC driver.

**Important**

After we update records using an UPDATE query, we can't undo the operation. If we want to know which records were updated, we first examine the results of a SELECT query that uses the same criteria, and then run the update query.

Maintain backup copies of data at all times. If we update the wrong records, we can retrieve them from our backup copies.

**Example**

This example changes values in the "Reports To" column to 5 for all employee records that currently have "Reports To" values of 2:

```
UPDATE [Employees] SET [Reports To] = 5 WHERE [Reports To] = 2;
```
Suppose you have a table named "Places" with "Name" and "Population" columns, and a table named "Births" with "Place," and "Births" columns, and you want to add "Births" from the "Births" table into the relevant record in the "Places" table.

To do this, use the following query:

```
UPDATE [Places] SET [Population] = [Population] + (SELECT Sum([Births]) FROM [Births] WHERE [Name] = [Place]);
```

**Tech Tip**

UPDATE queries support use of table name aliases.

**Clauses and Operators**

**ALL, ANY, SOME Quantifiers**

Determine if the value of an expression is equal to any or all values in a specified list.

**Syntax**

```
value comparison {ALL | ANY | SOME} list
```

The ALL, ANY, and SOME quantifiers have these parts:

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>An expression for the value of interest. This can be</td>
</tr>
<tr>
<td></td>
<td>either a numeric, date or string value, or a record.</td>
</tr>
<tr>
<td>comparison</td>
<td>A comparison operator (&lt;, &lt;=, &gt;, &gt;=, =, or &lt;&gt;).</td>
</tr>
<tr>
<td>list</td>
<td>A list of values or a query.</td>
</tr>
</tbody>
</table>

**Remarks**

Use the ALL quantifier to check if the value of interest is less than (or greater than, depending on the comparison operator) all values in a list.

Use the ANY or SOME quantifier, which are synonymous, to check if the value of interest is less than (or, say, greater than, depending on the comparison operator) any of the values in a list.

**Examples**

This example uses the ALL quantifier to select orders for which freight is greater than that of any order in 1994:

```
SELECT * FROM [Orders] WHERE [Freight] > ALL (SELECT [Freight] FROM [Orders] WHERE Year([Order Date]) = 1994);
```

This example uses the ANY quantifier to select orders for which freight is more than ten thousand times greater than that of any order in 1994 except those with zero freight. Note that unlike the above query, this query can not be reworked to use an aggregate function instead of a quantifier:

```
SELECT * FROM [Orders] WHERE [Freight] * 0.0001 > ANY (SELECT [Freight] FROM [Orders] WHERE Year([Order Date]) = 1994 AND [Freight] > 0);
```
This example uses the ANY quantifier to select all customers from France or Austria. The quantifier can be replaced with the IN operator:

```
SELECT * FROM [Customers] WHERE [Country] = ANY (VALUES ("France"), ("Austria"));
```

This example uses the ANY quantifier to select all employees who served at least one customer from the employee's country and city of origin:

```
SELECT * FROM [Employees] WHERE ([Country], [City]) = ANY (SELECT [Customers].[Country], [Customers].[City] FROM [Customers] INNER JOIN [Orders] ON [Orders].[Customer ID] = [Customers].[Customer ID] WHERE [Orders].[Employee ID] = [Employees].[Employee ID]);
```

See Also

ALL, DISTINCT, SKIP, TOP Quantifiers
ALL, DISTINCT, SKIP, TOP Quantifiers
Specify records returned by a query.

Syntax

See the SELECT statement.

Remarks

Use the ALL quantifier to return all records that meet the conditions in a query. If the query does not contain any quantifiers, the ALL quantifier is assumed. The following two examples are equivalent and return all records in the "Employees" table:

```
SELECT * FROM [Employees];
SELECT ALL * FROM [Employees];
```

Use the DISTINCT quantifier to omit records that contain duplicate data in the selected columns.

The DISTINCTROW quantifier is identical to the DISTINCT quantifier.

Use the TOP quantifier to return a certain number or a certain percentage of records that meet the conditions in a query. The TOP quantifier is typically used with an ORDER BY clause that orders records returned by the query by values in one or more columns.

Use the SKIP quantifier to skip a certain number or a certain percentage of records that meet the conditions in a query. Similarly to the TOP quantifier, the SKIP quantifier is typically used with an ORDER BY clause that orders records returned by the query by values in one or more columns. Using a combination of the SKIP and TOP quantifiers provides an easy way to "page" through query records.

The number or the percentage of records used by the TOP and SKIP quantifiers can be parameterized using the PARAMETERS declaration.

Examples

This example uses the DISTINCT quantifier to select all different titles of employees:

```
SELECT DISTINCT [Title] FROM [Employees];
```

This example uses the DISTINCT quantifier to select all types of discounts made on products (note that the DISTINCT quantifier leaves different discounts made on the same product as well as different products offered at the same discount):

```
SELECT DISTINCT [Product Name], [Discount] FROM [Products] INNER JOIN [Order Details] ON [Products].[Product ID] = [Order Details].[Product ID];
```

This example uses the TOP quantifier to select 5 products with highest prices:

```
SELECT TOP 5 * FROM [Products] ORDER BY [Unit Price] DESC;
```

This example uses the TOP quantifier to select 10% of products with the lowest amount of units in stock:

```
SELECT TOP 10 PERCENT * FROM [Products] ORDER BY [Units In Stock];
```

This example uses the SKIP quantifier to select all but the 5 products with highest prices:
SELECT SKIP 5 * FROM [Products] ORDER BY [Unit Price] DESC;

This example uses the SKIP and TOP quantifiers to select the second 10% of products (that is, between 10% and 20%) with the lowest amount of units in stock:

SELECT SKIP 10 PERCENT TOP 10 PERCENT * FROM [Products]
ORDER BY [Units In Stock];

This example uses a parameterized version of the TOP quantifier to select a user-specified number of products with the lowest amount of units in stock:

PARAMETERS [Number of Products] INTEGER;
SELECT TOP [Number of Products] * FROM [Products] ORDER BY [Units In Stock];

See Also

ALL, ANY, SOME Quantifiers
PARAMETERS Declaration
BETWEEN AND Operator
Determines whether the value of an expression falls within a specified range of values.

Syntax

value [NOT] BETWEEN value AND value

The BETWEEN AND operator syntax has these parts:

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>An expression for the value of interest, and expressions giving the bounds for this value. This can be either a numeric, date or string value, or a record.</td>
</tr>
</tbody>
</table>

Remarks

If the value of interest is greater than or equal to the first bound and less than or equal to the second bound, the operator returns True; otherwise, it returns False. Therefore, if the first bound is greater than the second bound, the operator will always return False. We can include the NOT operator to evaluate the opposite condition (that is, whether the value lies outside the range defined by the given bounds).

We can use the BETWEEN AND operator with numbers, strings, dates, but not binaries.

If any of the values is NULL, the BETWEEN AND operator also returns NULL.

We can use the BETWEEN AND operator with records that have the same number of columns of compatible types. The records are compared from the leftmost column to the rightmost column. We can also construct records using lists of values.

Examples

This example uses the BETWEEN AND operator to select all orders made between the 1st of June, 1991 and the 30th of June, 1991:

```
```

This example uses the BETWEEN AND operator to select all products with unit price between 10 and 14.95 dollars (inclusive):

```
SELECT * FROM [Products] WHERE [Unit Price] BETWEEN 10 AND 14.95;
```

This example uses the BETWEEN AND operator to select all customers whose name starts with a letter E, F, G, H, or I:

```
SELECT * FROM [Customers] WHERE UCase(Left([Contact Name], 1)) BETWEEN "E" AND "I";
```

This example uses the BETWEEN AND operator to select all orders in the 1992 between March and May:

```
SELECT * FROM [Orders] WHERE (Year([Order Date]), Month([Order Date])) BETWEEN (1992, 3) AND (1992, 5);
```
CASE Operator
Returns a value depending on a set of conditions.

Syntax

CASE value WHEN value THEN value [WHEN value THEN value ...] [ELSE value] END

or

CASE WHEN condition THEN value [WHEN condition THEN value ...] [ELSE value] END

The CASE operator has these parts:

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>The examined value, the example value, or the returned value.</td>
</tr>
<tr>
<td>condition</td>
<td>The examined Boolean expression.</td>
</tr>
</tbody>
</table>

Remarks

Use the first form of the CASE operator to translate a value to another value. A common use of the CASE operator is to replace codes or abbreviations with more readable values or vice versa. When used in the first form, the operator checks the examined value against the values in the WHEN clauses in the order they appear in the operator, and if the examined value is equal to that in a WHEN clause, the operator returns the value in the corresponding THEN clause. If the examined value is not equal to that in any WHEN clause, the operator returns the value in the ELSE clause, or NULL if the ELSE clause is omitted.

Use the second form of the CASE operator to evaluate one or more conditions and return a value depending on the results of the evaluation. When used in the second form, the operator evaluates the conditions in the WHEN clauses in the order they appear in the operator, and if a condition evaluates to True, returns the value in the corresponding THEN clause. If none of the conditions returns True, the operator returns the value in the ELSE clause, or NULL if the ELSE clause is omitted.

Examples

This example uses the CASE operator to report a home country for each employee, replacing "UK" with "United Kingdom," and "USA" with "United States of America":

```
SELECT [First Name], [Last Name], CASE [Country] WHEN "UK" THEN "United Kingdom" WHEN "USA" THEN "United States of America" ELSE [Country] END AS [Country] FROM [Employees];
```

This example uses the CASE operator to classify the order date of each order into seasons:

```
SELECT [Orders].*, CASE WHEN Month([Order Date]) BETWEEN 3 AND 5 THEN "Spring" WHEN Month([Order Date]) BETWEEN 6 AND 8 THEN "Summer" WHEN Month([Order Date]) BETWEEN 9 AND 11 THEN "Autumn" ELSE "Winter" END AS [Order Season] FROM [Orders];
```

This example uses the CASE operator to classify the freight of each order into "high," "medium," and "low" classes:

```
SELECT [Orders].*, CASE WHEN [Freight] >= 200 THEN "High" WHEN [Freight] >= 50 THEN "Medium" WHEN [Freight] > 0 THEN "Low" END AS [Freight Class] FROM [Orders];
```
CAST Operator
Explicitly casts the given value to another type.

Syntax

CAST(value AS type)

The CAST operator has these parts:

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>The value of interest.</td>
</tr>
<tr>
<td>type</td>
<td>The target type.</td>
</tr>
</tbody>
</table>

Remarks

Use the CAST operator to explicitly cast a value to another type. For the list of supported types, see the Data Types topic.

The CAST operator works like CDbl, CInt, CStr and other similar functions but allows using the explicit name of the type.

Examples

This example uses the CAST operator to select the rounded price of each product:

```
SELECT [Product Name], CAST([Unit Price] AS INT) AS [Rounded Price] FROM [Products];
```

This example uses the CAST operator and the IS NULL operator to select all employees with numeric postal codes:

```
SELECT * FROM [Employees] WHERE CAST([Postal Code] AS INT) IS NOT NULL;
```

We might use the above example in situations where we would like to differentiate between employees with numeric postal codes (as in the case of ZIP codes in the US, which contain only numbers) and those with postal codes that use both letters and numbers (as in the case of Canadian or UK postal codes).
CROSS JOIN Operator
Combines records from two tables without any conditions.

Syntax

table CROSS JOIN table

The CROSS JOIN operator has these parts:

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>table</td>
<td>The names of the tables from which records are combined.</td>
</tr>
</tbody>
</table>

Remarks

A CROSS JOIN operator is analogous to an INNER JOIN operator without a condition.

A CROSS JOIN operator is frequently used with a WHERE clause that filters the resulting records using criteria that would be not be used within an INNER JOIN.

A CROSS JOIN operator is also frequently used with the same table. In this case, it is often necessary to provide an alternate name for one of the copies of the table, so that the names of columns in the two copies of the table are no longer ambiguous.

Example

This example uses the CROSS JOIN operator and a WHERE clause to select the orders that were shipped later than those orders made after them (the DISTINCT operator collapses records for the same order into one record):

```sql
WHERE [Orders].[Order Date] < [Copy].[Order Date] AND [Orders].[Shipped Date] > [Copy].[Shipped Date];
```
EXCEPT Operator

Filters the results of one query or table so that it does not include the results of another query or table.

Syntax

```
query EXCEPT [ALL] query
```

The EXCEPT operator has these parts:

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>query</td>
<td>A table or a query that returns a table.</td>
</tr>
</tbody>
</table>

Remarks

We can use the EXCEPT operator to remove all records occurring in the second query or table from the results in the first query or table. The following example uses the EXCEPT operator to remove orders shipped after July 1, 1991 from orders ordered after June 1, 1991, to return all orders ordered and shipped in June, 1991:

```
```

We can chain several EXCEPT operators together to combine the results of several tables or queries. We can also chain EXCEPT operators with UNION and INTERSECT operators. The priority of the EXCEPT operator is lower than that of the INTERSECT operator, and is the same as that of the UNION operator.

By default, no duplicate records are returned when we use an EXCEPT operator; however, we can include the ALL predicate to ensure that all records are returned.

Both tables or queries used in an EXCEPT operator must have the same number of columns, which must be of compatible data types.

When using an EXCEPT operator on queries, use aliases only in the first SELECT statement because aliases used in any other SELECT statement but the first are ignored. In the ORDER BY clause, refer to columns by what they are called in the first table or query.

Examples

This example retrieves the names of all suppliers in USA who have a fax:

```
SELECT [Company Name], [Fax] FROM [Suppliers] WHERE [Country] = "USA" EXCEPT SELECT [Company Name], [Fax] FROM [Suppliers] WHERE [Fax] = ""
```

This example retrieves the names of all customers in France except those who live in Paris and those who ordered something in June, 1991:

```
```
**EXISTS Operator**
Determines if a given list of values contains at least one value.

**Syntax**

```sql
EXISTS list
```

The EXISTS operator has these parts:

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>A list of values or a query.</td>
</tr>
</tbody>
</table>

**Remarks**

Use the EXISTS operator to check if a list of values is empty or not. The EXISTS operator is typically used with a query, but we could use it with a list of values, in which case the operator will always return True.

**Examples**

This example uses the EXISTS operator to show whether each particular employee has served at least one order in July (any year):

```sql
SELECT [First Name], [Last Name], EXISTS (SELECT * FROM [Orders] WHERE [Orders].[Employee ID] = [Employees].[Employee ID] AND Month([Order Date]) = 7) FROM [Employees];
```

This example uses the EXISTS operator to select only those products that have a supplier in London (this can also be done with the INNER JOIN operator):

```sql
SELECT * FROM [Products] WHERE EXISTS (SELECT * FROM [Suppliers] WHERE [Suppliers].[Supplier ID] = [Products].[Supplier ID] AND [City] = "London");
```
FROM Clause
Specifies the tables or queries that contain the data for a SELECT statement.

Syntax

FROM query [AS] alias [(column [, column ...])] [, query ...]

The FROM clause has these parts:

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>query</td>
<td>A table or a query returning a table.</td>
</tr>
<tr>
<td>alias</td>
<td>An alternate name for a table or a query.</td>
</tr>
<tr>
<td>column</td>
<td>A column of a table or a query that is to be processed by the enclosing query.</td>
</tr>
</tbody>
</table>

Remarks

The order of the tables or queries in a FROM clause is not important.

We can specify an alternate name for each item in the FROM clause. The following example selects all cases of non-zero discounts on tofu using the "Order Details" and "Products" tables renaming the "Order Details" to "D," and "Products" to "P," to shorten the expressions in the INNER JOIN operator:

```sql
```

We can restrict the columns returned by each item in the FROM clause by supplying a list of desired columns after the item. The following example restricts data selected from the "Employees" table to only the first names and last names:

```sql
SELECT * FROM [Employees] ([First Name], [Last Name]);
```

Editing the column list to replace the asterisk with the names of the relevant columns would achieve the same result:

```sql
SELECT [First Name], [Last Name] FROM [Employees];
```

The difference between the above queries is that the first query filters the columns before they are available for the rest of the query, and the second query filters the columns after they have been processed by the rest of the query. Hence, the first of the modifications below will not work but the second will:

```sql
[Incorrect] SELECT * FROM [Employees] ([First Name], [Last Name]) WHERE [Employee ID] > 10;
```

```sql
[Correct] SELECT [First Name], [Last Name] FROM [Employees] WHERE [Employee ID] > 10;
```

Examples

This example selects the names and descriptions of all product categories using the FROM clause to restrict the data in the "Categories" table to only the desired columns:

```sql
SELECT * FROM [Categories] ([Category Name], [Description]);
```
This example selects orders to Caracas using the FROM clause to restrict the data in the "Orders" table to only the columns used by the query, and then using the column list to further restrict the data to only the output columns:

```
SELECT [Order Date], [Freight] FROM [Orders] ([Order Date], [Freight], [Ship City]) WHERE [Ship City] = "Caracas";
```
GROUP BY Clause
Combines records with identical values in the specified column or columns into a single record.

Syntax

GROUP BY expression [, expression ...]

The GROUP BY clause has these parts:

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>expression</td>
<td>The expression used for grouping. Can be a column name.</td>
</tr>
</tbody>
</table>

Remarks

Use the GROUP BY clause to group all records with the same value of one or more expressions into a single record.

We use the WHERE clause to exclude rows we do not want grouped, and we use the HAVING clause to filter records after they’ve been grouped.

We can not group records using binary columns.

When the GROUP BY clause is used, each item in the column list must either be exactly equal to one of the grouping expressions, or be an aggregate function, such as Sum or the Avg function.

Examples

This example uses the GROUP BY clause to create a list of unique job titles and the number of employees with each title:

```sql
SELECT [Title], Count(*) AS [Number of Employees] FROM [Employees] GROUP BY [Title];
```

This example uses the GROUP BY clause to compute the minimum, maximum, and total freight of orders for each customer:

```sql
SELECT [Customer ID], Min([Freight]) AS [Minimum], Max([Freight]) AS [Maximum], Sum([Freight]) AS [Total] FROM [Orders] GROUP BY [Customer ID];
```

This example is similar to the above, but uses the INNER JOIN operator to report the name of the customer instead of the customer ID:

```sql
SELECT [Company Name], Min([Freight]) AS [Minimum], Max([Freight]) AS [Maximum], Sum([Freight]) AS [Total] FROM [Orders] INNER JOIN [Customers] ON [Customers].[Customer ID] = [Orders].[Customer ID] GROUP BY [Company Name];
```

This example uses the GROUP BY clause to compute the number of orders shipped in one, two, three or other number of days:

```sql
SELECT DateDiff("d", [Shipped Date], [Order Date]) AS [Shipment Days], Count(*) AS [Number of Orders] FROM [Orders] GROUP BY DateDiff("d", [Shipped Date], [Order Date]);
```
HAVING Clause
Filters records grouped by the GROUP BY clause.

Syntax

HAVING condition

The HAVING clause has these parts:

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>condition</td>
<td>A Boolean expression that determines which grouped records to display.</td>
</tr>
</tbody>
</table>

Remarks

Use the HAVING clause to filter the records after they have been grouped by the GROUP BY clause. The HAVING clause is similar to the WHERE clause, which determines which records are selected, but the WHERE clause operates before the records are grouped, and the HAVING clause operates after that.

If we use the HAVING clause without the GROUP BY clause, the entire table is considered as a single group, and the resulting table has either one record or no records, depending on whether the HAVING clause returns True or False.

Examples

This example uses the GROUP BY clause, the HAVING clause, and the WHERE clause to select the job titles assigned to more than one employee in the Washington region. The WHERE clause restricts the employees to the Washington region, and the HAVING clause restricts the titles to those assigned to more than one employee:

```sql
SELECT [Title], Count(*) as [Total] FROM [Employees] WHERE [Region] = "WA"
GROUP BY [Title] HAVING Count(*) > 1;
```

This example uses the GROUP BY clause and the HAVING clause to select total freight per year for years 1993 and later:

```sql
SELECT Sum([Freight]) AS [Total], Year([Order Date]) AS [Year] FROM [Orders]
GROUP BY Year([Order Date]) HAVING Year([Order Date]) >= 1993;
```
**IN Operator**
Determines whether the value of an expression is equal to any of several values in a specified list.

**Syntax**

```
value [NOT] IN list
```

The In operator syntax has these parts:

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>An expression for the value of interest. This can be either a numeric, date or string value, or a record.</td>
</tr>
<tr>
<td>list</td>
<td>A list of values or a query.</td>
</tr>
</tbody>
</table>

**Remarks**

If the value of interest is found in the list of values, the operator returns True; otherwise, it returns False. We can include the NOT operator to evaluate the opposite condition (that is, whether the value is not in the list of values).

We can use the IN operator with numbers, strings, dates, but not binaries.

We can use the IN operator with records that have the same number of columns of compatible types. We can also construct records using lists of values.

**Examples**

This example uses the IN operator to locate all orders shipped to Lancashire and Essex:

```
SELECT [Shipped Date], [Ship Region] FROM [Orders] WHERE [Ship Region] IN ('Lancashire', 'Essex');
```

This example uses the IN operator with a sub-query to locate all orders handled by the personnel in London (the sub-query can be replaced with an INNER JOIN operator):

```
```

This example uses the IN operator and a sub-query to locate all orders shipped to Finland or Denmark by Federal Shipping (as in the example above, the sub-query can be replaced with an INNER JOIN operator):

```
SELECT * FROM [Orders] WHERE ([Ship Country], (SELECT [Company Name] FROM [Shippers] WHERE [Shipper ID] = [Ship Via])) IN ('Finland', 'Federal Shipping', 'Denmark', 'Federal Shipping');
```
INNER JOIN Operator
Combines records from two tables.

Syntax

table [INNER] JOIN table ON condition

The INNER JOIN operator has these parts:

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>table</td>
<td>The names of the tables from which records are combined.</td>
</tr>
<tr>
<td>condition</td>
<td>A Boolean expression restricting the records in the resulting table.</td>
</tr>
</tbody>
</table>

Remarks

We can use an INNER JOIN operator in any FROM clause. For example, we can use INNER JOIN with the "Categories" and "Products" tables to select all products in each category. In contrast, to select all categories (even if some categories have no products) or all products (even if some do not belong to any category), we need to use a LEFT JOIN or RIGHT JOIN operator to create an outer join.

The following example shows how we could join the "Categories" and "Products" tables using the "Category ID" column:

```
SELECT [Category Name], [Product Name] FROM [Categories] INNER JOIN [Products]
ON [Categories].[Category ID] = [Products].[Category ID];
```

We can use more than one column in a join condition.

We can nest JOIN statements using the following syntax:

table join (table join (table ...) ON condition) ON condition

Examples

This example uses the INNER JOIN operator to select all orders assigned to Andrew Fuller:

```
SELECT [Order ID], [Order Date] FROM [Orders] INNER JOIN [Employees]
ON [Employees].[Employee ID] = [Orders].[Employee ID] AND [Employees].[First Name]
= "Andrew" AND [Employees].[Last Name] = "Fuller";
```

The word "INNER" can be omitted, so the above query can be rewritten as:

```
SELECT [Order ID], [Order Date] FROM [Orders] JOIN [Employees]
ON [Employees].[Employee ID] = [Orders].[Employee ID] AND [Employees].[First Name]
= "Andrew" AND [Employees].[Last Name] = "Fuller";
```

This example uses nested INNER JOIN operators and GROUP BY clause to produce a list of employees and their total sales using "Employees," "Orders," and "Order Details" tables:

```
SELECT Sum([Unit Price] * [Quantity]) AS [Sales], ([First Name] & " " & [Last Name]) AS [Name] FROM [Employees] INNER JOIN ([Orders] INNER JOIN [Order Details] ON [Order Details].[Order ID] = [Orders].[Order ID])
ON [Orders].[Employee ID] = [Employees].[Employee ID]
GROUP BY [Name];
```
[Orders].[Employee ID] = [Employees].[Employee ID] GROUP BY ([First Name] & " " & [Last Name]);
INTERSECT Operator
Restricts the results of a query or table to only those results found in another query or table.

Syntax

query INTERSECT [ALL] query

The INTERSECT operator has these parts:

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>query</td>
<td>A table or a query that returns a table.</td>
</tr>
</tbody>
</table>

Remarks

We can use the INTERSECT operator to remove all records not found in the second query or table from the results of the first query or table. The following example uses the INTERSECT operator to restrict orders ordered after June 1, 1991 to only those orders shipped before July 1, 1991, to return all orders ordered and shipped in June, 1991:

```sql
```

We can chain several INTERSECT operators together to combine the results of several tables or queries. We can also chain INTERSECT operators with UNION and EXCEPT operators. The priority of the INTERSECT operator is higher than the priority of UNION and EXCEPT operators.

By default, no duplicate records are returned when we use an INTERSECT operator; however, we can include the ALL predicate to ensure that all records are returned.

Both tables or queries used in an INTERSECT operator must have the same number of columns, which should be of compatible data types.

When using an INTERSECT operator on queries, use aliases only in the first SELECT statement because aliases used in any other SELECT statement but the first are ignored. In the ORDER BY clause, refer to columns by what they are called in the first table or query.

Examples

This example retrieves the names of all employees in the US who were hired in 1992 or later:

```sql
SELECT [First Name], [Last Name] FROM [Employees] WHERE [Country] = "USA" INTERSECT SELECT [First Name], [Last Name] FROM [Employees] WHERE [Hire Date] >= #1/1/1992#;
```

This example retrieves the names of all customers in France (__) who placed orders in June, 1991, except those who live in Paris:

```sql
```

The parentheses around EXCEPT make the query engine first locate all customers from France, then remove those customers living in Paris, and then leave the customers who placed orders in June, 1991. Omitting the parentheses would make the query engine first locate all customers who live in Paris, then leave those customers who placed orders in June, and then subtract them from all the customers in France, which would obviously lead to a different result.
IS Operator
Checks whether a given Boolean value is True, False, or unknown.

Syntax

\[ \text{value IS [NOT] \{TRUE | FALSE | UNKNOWN\}} \]

The IS operator has these parts:

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>The value of interest</td>
</tr>
</tbody>
</table>

Remarks

Use the IS operator to check if the Boolean value is True, False, or unknown. The unknown value can occur as a combination of a True or False and a NULL, for example, True AND NULL is unknown. Note that some combinations of a True and a False and a NULL are not unknown, for example, True OR NULL is True.

The following table lists the logical rules used by the NOT operator:

<table>
<thead>
<tr>
<th>Expression</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOT True</td>
<td>False</td>
</tr>
<tr>
<td>NOT False</td>
<td>True</td>
</tr>
<tr>
<td>NOT Unknown</td>
<td>Unknown</td>
</tr>
</tbody>
</table>

The following table lists the logical rules used by the AND operator:

<table>
<thead>
<tr>
<th>AND True</th>
<th>AND False</th>
<th>AND Unknown</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>False</td>
<td>Unknown</td>
</tr>
<tr>
<td>False</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>Unknown</td>
<td>False</td>
<td>Unknown</td>
</tr>
</tbody>
</table>

For example, False AND Unknown results in False.

The following table lists the logical rules used by the OR operator:

<table>
<thead>
<tr>
<th>OR True</th>
<th>OR False</th>
<th>OR Unknown</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>Unknown</td>
</tr>
<tr>
<td>Unknown</td>
<td>Unknown</td>
<td>Unknown</td>
</tr>
</tbody>
</table>

For example, False OR Unknown results in Unknown.

The following table lists the logical rules used by the IS operator:

<table>
<thead>
<tr>
<th>IS True</th>
<th>IS False</th>
<th>IS Unknown</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
For example, `False IS Unknown` results in `FALSE`.

Use the IS NULL operator to check if a value (not necessarily Boolean) is NULL or not.

**Examples**

This example uses the IS operator to select all employees whose birthday falls on August:

```
SELECT * FROM [Employees] WHERE (Month([Birth Date]) = 8) IS TRUE;
```

The **IS TRUE** can always be omitted, so the above example could be rewritten as:

```
SELECT * FROM [Employees] WHERE Month([Birth Date]) = 8;
```

This example uses the IS operator to select all orders for which the order date and required date are not in the same month:

```
SELECT * FROM [Orders] WHERE (Month([Order Date]) = Month([Required Date])) IS FALSE;
```

This example uses the IS operator and a subquery to select all orders to the United Kingdom that do not include chocolate:

```
SELECT * FROM [Orders] WHERE ([Ship Country] = "UK" AND (SELECT Sum([Quantity])
```
IS NULL Operator
Checks whether the given value is NULL or not.

Syntax

value IS [NOT] NULL

The IS NULL operator has these parts:

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>The value of interest.</td>
</tr>
</tbody>
</table>

Remarks

Use the IS NULL operator to check whether a value is NULL or not. The IS NULL operator is typically used with a query that returns a single record. In this case, the operator returns True if all of the values within the returned record are NULL and False otherwise. If the query returns more than one record the IS NULL operator returns True (not False!).

Examples

This example uses the IS NULL operator to select orders that were the first orders processed in that month (this also can be done with the EXISTS or INNER JOIN operators or with the Count aggregate function):

```sql
SELECT * FROM [Orders] WHERE (SELECT Min([Copy].[Order Date]) FROM [Orders] AS [Copy] WHERE [Copy].[Order Date] < [Orders].[Order Date] AND Year([Copy].[Order Date]) = Year([Orders].[Order Date]) AND Month([Copy].[Order Date]) = Month([Orders].[Order Date])) IS NULL;
```

This example uses the IS NULL operator to select employees who have processed orders on their birthday (this also can be done with the EXISTS or INNER JOIN operators or with the Count aggregate function):

```sql
SELECT * FROM [Employees] WHERE (SELECT Min([Orders].[Order ID]) FROM [Orders] WHERE [Orders].[Employee ID] = [Employees].[Employee ID] AND Month([Order Date]) = Month([Birth Date]) AND Day([Order Date]) = Day([Birth Date])) IS NOT NULL;
```
LEAVING Clause
Filters records exploded by the SPLIT BY clause.

Syntax

LEAVING condition

The LEAVING clause has these parts:

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>condition</td>
<td>A Boolean expression that determines which exploded records to display.</td>
</tr>
</tbody>
</table>

Remarks

Use the LEAVING clause to filter the records after they have been exploded by the SPLIT BY clause. The LEAVING clause is similar to the WHERE clause, which determines which records are selected, but the WHERE clause operates before the records are exploded and the LEAVING clause operates after that.

The LEAVING clause cannot be used without the SPLIT BY clause.

Examples

This example uses the SPLIT BY clause and the LEAVING clause to select all islands in areas in drawing D the area of which exceeds 10 miles:

```
SELECT [I], Area([I], "mi") FROM [D]
SPLIT BY Islands([ID]) AS [I]
LEAVING Area([I], "mi") > 10;
```

This example uses the SPLIT BY clause and the LEAVING clause to select all coordinates in objects in drawing D the X coordinates of which are between -1000 and 1000:

```
SELECT [P] FROM [D]
SPLIT BY Coords([ID]) AS [P]
LEAVING CentroidX([P]) BETWEEN -1000 AND 1000;
```

This example uses the SPLIT BY clause and the LEAVING clause to select all coordinates in objects in drawing D which are within 1 mile of the object centroid:

```
SELECT [P] FROM [D]
SPLIT BY Coords([ID]) AS [P]
LEAVING Distance([P], Centroid([ID]), "mi") < 1;
```

See Also

SPLIT BY Clause
WHERE Clause
LIKE Operator
Checks if a string value matches a given pattern.

Syntax

expression [NOT] LIKE expression

The LIKE operator has these parts:

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>expression</td>
<td>The value of interest and a pattern.</td>
</tr>
</tbody>
</table>

Remarks

Use the LIKE operator to check if a string value matches a given pattern. If a value matches a pattern, the operator returns True; otherwise, it returns False.

In a pattern, use the underscore character (_) to match any single character. Use the percent sign (%) to match any number of characters including zero. Use the backslash character and the underscore character (\_) to match the underscore character, use the backslash character and the percent sign (\%) to match the percent sign, use two backslash characters (\\) to match the backslash character. Use any other character, like a letter or a digit, to match itself.

The LIKE operator is case-sensitive. To match values without case, convert both the string and the pattern to either upper or lower case with UCase or LCase.

If either the checked value or the pattern is NULL, the LIKE operator also returns NULL.

Examples

The following table shows sample expressions using the LIKE operator and the value returned by each expression:

<table>
<thead>
<tr>
<th>String Expression</th>
<th>Returns</th>
</tr>
</thead>
<tbody>
<tr>
<td>abbc LIKE a%</td>
<td>TRUE</td>
</tr>
<tr>
<td>abbc LIKE _c</td>
<td>FALSE</td>
</tr>
<tr>
<td>abbc LIKE %c</td>
<td>TRUE</td>
</tr>
<tr>
<td>abbc LIKE %bb</td>
<td>FALSE</td>
</tr>
<tr>
<td>a_bc LIKE %</td>
<td>TRUE</td>
</tr>
<tr>
<td>a_bc LIKE a_bc</td>
<td>TRUE (because _ matches any character)</td>
</tr>
<tr>
<td>a_bc LIKE a_bc</td>
<td>TRUE (because \ matches _)</td>
</tr>
<tr>
<td>a_bc LIKE a_c</td>
<td>TRUE</td>
</tr>
<tr>
<td>ab\c LIKE %%</td>
<td>FALSE</td>
</tr>
<tr>
<td>ab\c LIKE %%</td>
<td>TRUE</td>
</tr>
</tbody>
</table>

This example uses the LIKE operator to select all products the name of which starts with "C":

    SELECT * FROM [Products] WHERE [Product Name] LIKE "C%";
This example uses the LIKE operator to select all products the name of which ends with "n":

```sql
SELECT * FROM [Products] WHERE [Product Name] LIKE "%n";
```

This example uses the LIKE operator to select all customers whose contact title includes the word "sales" (ignoring case):

```sql
SELECT * FROM [Customers] WHERE LCase([Contact Title]) LIKE "%sales%";
```

This example uses the LIKE operator to select all customers whose phone number contains a non-empty area code in parentheses:

```sql
SELECT * FROM [Customers] WHERE [Phone] LIKE "%(_%)%";
```

This example uses the LIKE operator and the PARAMETERS declaration to locate all customers whose company name matches a pattern supplied as a parameter:

```sql
PARAMETERS [Pattern] TEXT;
SELECT * FROM [Customers] WHERE [Company Name] LIKE [Pattern];
```
**LIKEX Operator**
Checks if a string value matches a given pattern using regular expressions.

**Syntax**

```
expression [NOT] LIKEX expression
```

The LIKEX operator syntax has these parts:

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>expression</td>
<td>The value of interest and a pattern.</td>
</tr>
</tbody>
</table>

**Remarks**

Use the LIKEX operator to check if a string value matches a given pattern. If the value matches the pattern, the operator returns True; otherwise, it returns False.

For syntax that may be used in the pattern see the Regular Expressions topic.

If either the checked value or the pattern is NULL, the LIKEX operator also returns NULL.

**Examples**

This example uses the LIKEX operator to select all products whose name contains "'s" at the end of the word that is not the last word:

```
SELECT * FROM [Products] WHERE [Product Name] LIKEX ".+'s .+";
```

This example uses the LIKEX operator to select all customers whose phone contains three-digit area code:

```
SELECT * FROM [Customers] WHERE [Phone] LIKEX ".*\(\d{3}\).*";
```

This example uses the LIKEX operator to select all customers whose company names consist of at least three words:

```
SELECT * FROM [Customers] WHERE [Company Name] LIKEX "\S* \S* \S.*";
```

This example uses the LIKEX operator to select all customers whose company name includes the word "the", ignoring the case:

```
SELECT * FROM [Customers] WHERE LCase([Company Name]) LIKEX ".*\bthe\b.*";
```
OPTIONS Clause

The OPTIONS clause allows a declaration of options for query execution and usage. At the present time, the only option for this clause is the CoordSys query function. The OPTIONS clause must be the first clause in the query, since it potentially specifies the operation of all the rest of the query.

Syntax

OPTIONS expression[, expression];

The OPTIONS clause has these parts:

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>expression</td>
<td>An expression using the CoordSys or Precision subclauses.</td>
</tr>
</tbody>
</table>

Remarks

In Manifold, we can use a query to produce temporary results which can be treated as a drawing, image or surface. In many cases, the coordinate system and other parameters of the component imported or linked from such a query will be set to default values. The OPTIONS clause allows specifying the values of these parameters to be different from default values.

The OPTIONS clause supports the following subclauses:

- **CoordSys** - Allows specifying the coordinate system of the component represented by the query. The syntax of the CoordSys subclause is similar to that of the CoordSys function specified in the Spatial Extensions topic. The CoordSys subclause can accept a string value with the name of the coordinate system preset, or a string value with the name of the component to take the coordinate system from, followed by the AS COMPONENT construct.

- **Precision** - Allows specifying the value of the location precision parameter of the drawing represented by the query. The Precision subclause can accept a floating-point value with the value of the location precision parameter in native units, or a string value with the name of the component to take the value of the location precision parameter from, followed by the AS COMPONENT construct.

The CoordSys subclause is especially useful with queries representing images and surfaces, which typically use virtual tables for existing images or surfaces.

Consider the following two queries operating on a surface named Surface:

```
SELECT * FROM [Surface] WHERE [Height (I)] > 100;

...and...

OPTIONS CoordSys("Surface" AS COMPONENT);
SELECT * FROM [Surface] WHERE [Height (I)] > 100;
```

Both queries select all pixels with a height greater than 100. However, linking a surface from the first query will create a surface in the default Orthographic projection, since pixels do not carry coordinate system data, but linking a surface from the second query will create a surface using the coordinate system (that is, the projection) of the original surface.

The following query selects all areas (but not points or lines) from the drawing named "Drawing". Linking that query as a drawing will set the value of the location precision parameter in the new drawing to that in "Drawing":

```
OPTIONS Precision("Drawing" AS COMPONENT);
SELECT * FROM [Drawing] WHERE CStr([Type (I)]) = "Area";
```
**Example**

We have a large surface and we would like to create a query that grabs only a portion of that surface within a given latitude and longitude range. We will then create a linked surface based on the query. The result will be a surface that is only a portion of the larger surface. We will use an OPTIONS clause in the query to make sure the resulting linked surface is correctly georegistered using the same projection as the original surface. This is a common technique for Internet Map Server (IMS) applications.

We begin with a surface showing dropout rates from high school in the US (blue represents lower dropout rates, red shows higher dropout rates). This is the same surface created in the Displaying Data in a Gradient Map example. We have colored the background black and have turned on a graticule showing latitude and longitude lines every five degrees.

We write a query called **Surface Query** that selects pixels between -85 and -80 degrees longitude and between 35 and 40 degrees latitude. The OPTIONS clause will ensure that linked surface created from the query will use the same projection as the original surface.

```
OPTIONS CoordSys("Surface" AS COMPONENT);
SELECT * FROM [Surface]
WHERE ([Longitude (I)] >= -85 AND [Longitude (I)] <= -80)
   AND ([Latitude (I)] >= 35 AND [Latitude (I)] <= 40);
```

If we create a linked surface from the query we see that only that portion of the surface within the desired latitude and longitude ranges is in the linked surface.
For use in an application like an IMS website, we would probably rewrite the query used in this example as a parameter query as seen above. This would allow users to specify the minimum and maximum longitude and latitude range to get the desired portion of the surface.
**ORDER BY Clause**

Sorts records using specified criteria in ascending or descending order.

**Syntax**

```
ORDER BY expression [ASC | DESC] [, expression [ASC | DESC] ...]
```

The ORDER BY clause has these parts:

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>expression</td>
<td>An expression with which to sort records.</td>
</tr>
</tbody>
</table>

**Remarks**

Use the ORDER BY clause to sort records using one or more expressions. The expressions can be columns, operations on columns, or literals. If the ORDER BY clause includes more than one expression, the records are sorted by the value returned by the first expression, then the records with the same value of the first expression are sorted by the value returned by the second expression, and so on.

Use a numeric expression that is a single number to sort by the value of an output column with the given index (one-based).

Use the ASC and DESC keywords to specify whether the sorting is done in ascending (ASC) or descending (DESC) order. If neither ASC nor DESC is specified, the sorting is done in ascending order.

We can sort records using numeric, date, or string expressions, but not binary expressions.

**Examples**

This example uses the ORDER BY clause to sort employees by last name in ascending order:

```
SELECT * FROM [Employees] ORDER BY [Last Name] ASC;
```

The ASC keyword is optional, so we may rewrite the above example as:

```
SELECT * FROM [Employees] ORDER BY [Last Name];
```

This example uses the ORDER BY clause to sort employees by country, and then by city:

```
SELECT * FROM [Employees] ORDER BY [Country], [City];
```

This example uses the ORDER BY clause and the INNER JOIN operator to sort orders by shipper in ascending order, and then by the number of days it took to ship the order in descending order:

```
SELECT [Company Name], [Order Date], [Shipped Date] FROM [Orders] INNER JOIN [Shippers] ON [Shipper ID] = [Ship Via] ORDER BY [Company Name], DateDiff("d", [Shipped Date], [Order Date]) DESC;
```

This example is similar to the example above, but includes the number of days it took to ship the order as the output column, and uses references to the output columns in the ORDER BY clause:

```
SELECT [Company Name], [Order Date], [Shipped Date], DateDiff("d", [Shipped Date], [Order Date]) AS [Shipment Days] FROM [Orders] INNER JOIN [Shippers] ON [Shipper ID] = [Ship Via] ORDER BY 1, 4 DESC;
```
OUTER (LEFT, RIGHT, FULL) JOIN Operators

Combine records from two tables allowing NULL values on left, right, or both sides.

Syntax

```
table {LEFT | RIGHT | FULL} [OUTER] JOIN table ON condition
```

The LEFT JOIN, RIGHT JOIN, and FULL JOIN operators have these parts:

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>table</td>
<td>The names of the tables from which records are combined.</td>
</tr>
<tr>
<td>condition</td>
<td>A Boolean expression restricting the records in the resulting table.</td>
</tr>
</tbody>
</table>

Remarks

Use a LEFT JOIN operator to create a left outer join. Left outer joins include all of the records from the first (left) of two tables, even if there are no matching values for records in the second (right) table.

Use a RIGHT JOIN operator to create a right outer join. Right outer joins include all of the records from the second (right) of two tables, even if there are no matching values for records in the first (left) table.

Use a FULL JOIN operator to create a full outer join. Full outer joins include all of the records from both tables, even if there are no matching values for records in the other table.

The following example shows how we could join the "Categories" and "Products" tables using the "Category ID" column to produce a list of all categories, including those that contain no products. The GROUP BY clause and the First aggregate function are used to collapse multiple records for products sharing the same category:

```
SELECT [Category Name], First([Product Name]) FROM [Categories] LEFT JOIN [Products] ON [Categories].[Category ID] = [Products].[Category ID] GROUP BY [Category Name];
```

The following example shows how we could join the "Categories" and "Products" tables using the "Category ID" column to produce a list of all products, including those that contain no categories:

```
SELECT [Category Name], [Product Name] FROM [Categories] RIGHT JOIN [Products] ON [Categories].[Category ID] = [Products].[Category ID];
```

To combine records disallowing NULL values on any side of a join, use an INNER JOIN operator.

We can use more than one column in a join condition.

When nesting any of the LEFT JOIN, RIGHT JOIN, and FULL JOIN operators with one or more of the INNER JOIN operators, for performance purposes it is recommended that we put LEFT JOIN, RIGHT JOIN, and FULL JOIN operators inside the INNER JOIN operators.

Examples

This example uses the LEFT JOIN operator and the GROUP BY clause to select employees and the number of orders they processed in June, 1991:

```
```
This example uses the LEFT JOIN operator, several INNER JOIN operators, and the GROUP BY clause to show whether a given product has or has not been processed at least once by Andrew Fuller:

SELECT [Product Name], "Yes" AS [Served] FROM [Products] LEFT JOIN ([Order Details] INNER JOIN ([Orders] INNER JOIN [Employees] ON [Orders].[Employee ID] = [Employees].[Employee ID] AND [First Name] = "Andrew" AND [Last Name] = "Fuller") ON [Order Details].[Order ID] = [Orders].[Order ID]) ON [Order Details].[Product ID] = [Products].[Product ID] GROUP BY [Product Name];
PARAMETERS Declaration
Declares the name and data type of each parameter in a parameter query.

Syntax

    PARAMETERS name type [(size)] [, name type [(size)] ...]

The PARAMETERS declaration has these parts:

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>The name of the parameter. Use brackets ([ ]) to enclose text that contains spaces or punctuation. For example, [Low price] and [Begin report with which month?] are valid name arguments.</td>
</tr>
<tr>
<td>type</td>
<td>The type of the parameter.</td>
</tr>
<tr>
<td>size</td>
<td>The size of the parameter in characters or bytes for text or binary parameters.</td>
</tr>
</tbody>
</table>

Remarks

For queries that we run regularly, we can use a PARAMETERS declaration to create a parameter query. A parameter query can help automate the process of changing query criteria. With a parameter query, our code will need to provide the parameters each time the query is run.

The PARAMETERS declaration is optional but when included precedes any other statement.

We can use the PARAMETERS declaration with SELECT, SELECT INTO, INSERT INTO, UPDATE, DELETE, and TRANSFORM statements.

We can include more than one parameter, separating them with commas. The following example includes two parameters:

    PARAMETERS [Low price] Currency, [Beginning date] DateTime;

We can use the parameters to substitute any value within the query. We cannot, however, use the parameters to substitute names of columns or tables, or to specify functions or operators. The following example expects two parameters to be provided and then applies the criteria to records in the "Orders" table:

    PARAMETERS [Low Freight] Double, [Beginning Date] DateTime;
    SELECT [Order ID], [Order Date], [Freight] FROM [Orders] WHERE [Freight] > [Low Freight] AND [Order Date] >= [Beginning Date];

Examples

This example requires the user to provide a job title and then selects all employees with the entered title:

    PARAMETERS [Employee Title] TEXT;
    SELECT [First Name], [Last Name], [Title] FROM [Employees] WHERE [Title] = [Employee Title];

This example requires the user to provide the first characters of the employee name and then selects all employees whose first or last name starts with the entered characters:

    PARAMETERS [Name Prefix] TEXT;
SELECT [First Name], [Last Name] FROM [Employees] WHERE [First Name] LIKE [Name Prefix] & "%" OR [Last Name] LIKE [Name Prefix] & "%";
POSITION Operator
Returns the position of a string value within another string value.

Syntax

POSITION(value IN value)

The POSITION operator has these parts:

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>The value of interest and the containing value.</td>
</tr>
</tbody>
</table>

Remarks

Use the POSITION operator to locate a string within another string. If the second value contains a substring that is equal to the first value the operator returns the starting location of that substring (one-based). If the second value contains several substrings equal to the first value the operator returns the starting location of the first such substring. If the second value contains no substrings equal to the first value, the operator returns 0.

The POSITION operator is case-sensitive.

Examples

This example uses the POSITION operator to select all products that typically ship in bottles:

```
SELECT * FROM [Products] WHERE POSITION("bottles" IN [Quantity Per Unit]) > 0;
```

This example uses the POSITION operator and the CAST operator to select all products the price of which is not an integer value of dollars (that is, which contains a decimal point in the price):

```
SELECT * FROM [Products] WHERE POSITION("." IN CAST([Unit Price] AS TEXT)) > 0;
```
SPLIT BY Clause
Explores records into groups of records using values in one or more columns.

Syntax

```
SPLIT BY expression[, expression ...]
```

The SPLIT BY clause has these parts:

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>expression</td>
<td>The expression used for splitting. Must contain a call to one of the split functions or be a query.</td>
</tr>
</tbody>
</table>

Remarks

Use the SPLIT BY clause to replace each record with a group of records using one or more splitting expressions. Each splitting expression must be a query or must contain a call to one of the special split functions used to convert a single value into a set of values.

When the SPLIT BY clause includes more than one splitting expression, the records are first split with the first expression, then the result is split with the second expression, and so on in left-to-right order.

The effect of the SPLIT BY clause is the reverse of the GROUP BY clause, in that the GROUP BY clause collapses groups of records into single records, and the SPLIT BY clause explodes single records into groups of records.

It is possible to use both the SPLIT BY clause and the GROUP BY clause in the same query. In this case, the SPLIT BY clause will be applied before the GROUP BY clause.

To filter the records returned by the SPLIT BY clause, use the LEAVING clause.

Examples

This example uses the SPLIT BY clause and a query to join each area in drawing D to each point in the same drawing (one could also do this using the INNER JOIN operator):

```sql
SELECT [ID] AS [AID], [PID] FROM [D] AS [DA]
WHERE IsArea([ID]) SPLIT BY
WHERE IsPoint([DP].[ID]) AND Contains([DA].[ID], [DP].[ID]));
```

This example uses the SPLIT BY clause to output each branch of each object in drawing D as a separate geom:

```sql
SELECT [ID], [P] FROM [D]
SPLIT BY Branches([ID]) AS [P];
```

This example uses the SPLIT BY clause to create the convex hull of each area in drawing D and to output the coordinates in the hulls as separate geoms:

```sql
SELECT [ID], [P] FROM [D]
WHERE IsArea([ID])
SPLIT BY Coords(ConvexHull([ID])) AS [P];
```

This example uses the SPLIT BY clause to create a Voronoi diagram of all points in drawing D and output a Voronoi area for each point:
SELECT [P] FROM (SELECT Voronoi(AllCoords([ID]))
AS [V] FROM [D]
WHERE IsPoint([ID]))
SPLIT BY Branches([V]) AS [P];

This example is similar to the above, but uses the INNER JOIN operator to only output a Voronoi area for each point the ID of which exceeds 5:

```
SELECT [ID], [P] FROM
(SELECT [P] FROM
(SELECT Voronoi(AllCoords([ID])) AS [V] FROM [D]
WHERE IsPoint([ID]))
SPLIT BY Branches([V]) AS [P])
INNER JOIN [D] ON Contains([P], [ID])
WHERE IsPoint([ID]) AND [ID] > 5;
```

This example uses the SPLIT BY clause and the GROUP BY clause to create a table of all possible values of X in all coordinates in the drawing D, along with their frequencies:

```
SELECT CentroidX([P]), Count(*) FROM [D]
SPLIT BY Coords([ID]) AS [P]
GROUP BY CentroidX([P]);
```

**See Also**

- **GROUP BY Clause**
- **LEAVING Clause**
SUBSTRING Operator

Returns a substring from a string value.

Syntax

```
SUBSTRING(value FROM start [FOR length])
```

The SUBSTRING operator has these parts:

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>The value of interest.</td>
</tr>
<tr>
<td>start</td>
<td>The starting location of the substring (one-based).</td>
</tr>
<tr>
<td>length</td>
<td>The length of the substring.</td>
</tr>
</tbody>
</table>

Remarks

Use the SUBSTRING operator to extract a substring from a string value. If the starting location or the length is less than or equal to zero the operator returns NULL. If the length parameter is omitted it is assumed to be equal to the remaining length of the string.

The SUBSTRING operator is case-sensitive.

Examples

This example uses the SUBSTRING operator to select the initials of each employee:

```
SELECT [First Name], [Last Name], SUBSTRING([First Name] FROM 1 FOR 1) & SUBSTRING([Last Name] FROM 1 FOR 1) AS [Initials] FROM [Employees];
```

This example uses the SUBSTRING operator and the POSITION operator to select the first word in the name of each product:

```
SELECT [Product Name], SUBSTRING([Product Name] FROM 1 FOR POSITION(" " IN [Product Name] & " ") - 1) AS [First Word] FROM [Products];
```
TRIM Operator
Trims characters from the left, right, or both sides of a string value.

Syntax

TRIM([[LEADING | TRAILING | BOTH] [value] FROM] value)

The TRIM operator has these parts:

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>A set of characters to trim and a string value.</td>
</tr>
</tbody>
</table>

Remarks

Use the TRIM operator to trim characters from the left, right, or both sides of a string value. If the set of characters is omitted the operator trims white space characters. If no LEADING, TRAILING, or BOTH keywords are specified the operator trims both sides of the string value.

The TRIM operator is case-sensitive.

Examples

This example uses the TRIM operator to trim the building number from the address of each customer:

```sql
SELECT [Address], TRIM(TRIM("1234567890, " FROM [Address])) AS [Trimmed Address] FROM [Customers];
```

This example uses the TRIM operator to remove the first word from the title of each employee (using the English alphabet assuming the data is in English):

```sql
SELECT [Title], TRIM(TRIM(LEADING "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz" FROM [Title])) AS [Trimmed Title] FROM [Employees];
```
UNION Operator

Merges the results of one query or table with another query or table.

Syntax

query UNION [ALL] query

The UNION operation has these parts:

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>query</td>
<td>A table or a query that returns a table.</td>
</tr>
</tbody>
</table>

Remarks

We can use the UNION operator to merge together the results of two tables or queries. The following example uses the UNION operator to merge two SELECT statements that select data from the "Orders" table:

```
```

We can chain several UNION operators together to merge the results of several tables or queries. We can also chain UNION operators with EXCEPT and INTERSECT operators. The priority of the UNION operator is lower than that of the INTERSECT operator, and is the same as the priority of the EXCEPT operator.

By default, no duplicate records are returned when we use a UNION operator; however, we can include the ALL predicate to ensure that all records are returned.

Both tables or queries used in a UNION operator must have the same number of columns, which must be of compatible data types.

When using a UNION operator on queries, use aliases only in the first SELECT statement because aliases used in any other SELECT statement but the first are ignored. In the ORDER BY clause, refer to columns by what they are called in the first table or query.

Examples

This example retrieves the names and cities of all suppliers and customers in Brazil:

```
SELECT [Company Name], [City] FROM [Suppliers] WHERE [Country] = "Brazil" UNION SELECT [Company Name], [City] FROM [Customers] WHERE [Country] = "Brazil";
```

This example retrieves the names of all customers who either live in Brazil or whose contact is a sales manager without duplicates:

```
SELECT [Contact Name], "Lives in Brazil" AS [Reason] FROM [Customers] WHERE [Country] = "Brazil" UNION SELECT [Contact Name], "Is a Sales Manager" FROM [Customers] WHERE [Contact Title] = "Sales Manager";
```
**UNIQUE Operator**
Determines whether the given list of values contains duplicate values.

**Syntax**

```sql
UNIQUE list
```

The UNIQUE operator has these parts:

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>A list of values or a query.</td>
</tr>
</tbody>
</table>

**Remarks**

Use the UNIQUE operator to check if a list of values contains duplicate values or not. If the list of values contains duplicates, the operator returns False; if the list of values does not contain duplicates, or is empty, the operator returns True. The UNIQUE operator is typically used with a query, but we could use it with a list of values.

**Examples**

This example uses the UNIQUE operator to show whether each particular employee has served the same customer more than once:

```sql
SELECT [First Name], [Last Name], UNIQUE (SELECT [Company Name] FROM [Customers] INNER JOIN [Orders] ON [Orders].[Customer ID] = [Customers].[Customer ID] WHERE [Orders].[Employee ID] = [Employees].[Employee ID]) FROM [Employees];
```

This example uses the UNIQUE operator to select only those products that were ordered only once by each customer:

```sql
```
WHERE Clause
Filters records selected by a query.

Syntax

WHERE condition

The WHERE clause has these parts:

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>condition</td>
<td>A Boolean expression that determines which records to display.</td>
</tr>
</tbody>
</table>

Remarks

Use the WHERE clause to filter records returned by a query. If the condition in the WHERE clause returns True, the record is included in the output; if it returns False, the record is excluded from the output. If the condition in the WHERE clause returns NULL, the record is excluded from the output.

In a query with a GROUP BY clause, the WHERE clause is similar to the HAVING clause in that it determines which records are selected, but the WHERE clause operates before the records are grouped, and the HAVING clause operates after records are grouped.

Examples

This example uses the WHERE clause to select employees whose last name is "King":

```
SELECT * FROM [Employees] WHERE [Last Name] = "King";
```

This example uses the WHERE clause to select orders made in July, 1992:

```
SELECT * FROM [Orders] WHERE Month([Order Date]) = 7 AND Year([Order Date]) = 1992;
```

This example uses the WHERE clause to select orders made in June or July, 1992:

```
SELECT * FROM [Orders] WHERE Month([Order Date]) IN (6, 7) AND Year([Order Date]) = 1992;
```

This example uses the WHERE clause to select orders made after June 1, 1992, that must be shipped before August 1, 1992, and that must ship to a country the name of which starts with either "F" or "U":

```
```

This example uses the WHERE clause to select orders to the UK the freight for which is greater than the average freight of all orders:

```
```
Examples

This section of the documentation sets forth step-by-step examples to help you learn Manifold. The examples in this Help system are aimed at elementary use of Manifold. We strongly urge you to repeat the examples by "following along" within Manifold.

Begin with the My First Map example and then read the other example topics in sequence. All the example topics are listed in the Examples chapter in the Help system. Get a table of contents in Help by clicking on the Show button on the Help toolbar if the tabs are hidden and then choosing the Contents tab. The Web version of this documentation might not be structured the same as the Microsoft Help version built into the installation. If you jump directly into a topic within web version you might not see the contents listing. You can always find the full web version of the documentation by going to the manifold.net web site and clicking on the User Manual link.

In addition to the step-by-step examples in the Examples chapter there are numerous small examples included throughout the rest of the documentation. The Other Examples topic provides a useful summary of the most important of these.

Visit the www.manifold.net web site from time to time and drill down into the Support pages there to check on new documentation that may be published. Advanced examples, tech support notes, scripting examples, case studies and other supplementary materials may be published on the manifold.net web site.

Files used in the Examples

Files used in the examples may be downloaded from the Product Downloads page on the manifold.net web site.

Important Configuration Notice

Very Important: Before starting Manifold, go to the Windows Control Panel Display dialog Effects tab and make sure that the Show window contents while dragging check box is not checked.

- In Windows XP this option is found in the Control Panel's Appearances and Themes - Display choice under the Appearance tab by pressing the Effects button.
- In Windows XP or Windows 2003, from the Start button open the Control panel and then open the Display dialog. Click on the Appearances tab and then press the Effects button. Uncheck the Show window contents while dragging check box.
- In Windows 2000, Windows ME and Windows 98 open the Control panel and then open the Display dialog. Click on the Effects tab, and uncheck the Show window contents while dragging check box.
- In Windows Vista, open the Control Panel and click Appearance and Personalization. Click Customize colors, then click Open classic appearance properties for more color options and click the Effects button. Uncheck the Show window contents while dragging check box.

Checking this box will greatly slow down the system when displaying complex maps and images because it forces a refresh of the window contents with each minor change in mouse position while dragging the window. Make sure the box is unchecked.

A Note for Access Users

Microsoft Access users may be familiar with the Northwind.mdb sample database provided by Microsoft with Access. The Nwind.mdb sample database provided with Manifold is a slightly earlier version of Northwind.mdb that Microsoft has authorized for redistribution. Nwind.mdb is for all practical purposes identical to the Northwind.mdb sample database distributed with Access. Access users should feel free to use either Nwind.mdb provided by Manifold or Northwind.mdb installed with Access to follow along in the examples. To avoid confusion, both versions of this database are referred to in these examples as the "Northwind" sample database.

Important Note when Using 64-bit Manifold Editions
If you have installed 64-bit Manifold, do the examples in 32-bit mode by launching Manifold System using the Manifold System (32-bit) shortcut. Some of the examples use file formats (noted below) that are not available for use in 64-bit Windows systems so they cannot be performed using Manifold in 64-bit mode. Launching Manifold in 32-bit mode will make it possible to follow along in the examples while learning Manifold.

Due to a lack of required Microsoft facilities in 64-bit Windows systems, Manifold in 64-bit mode cannot export, import, export or link to DB, HTML, MDB, XLS or WKx format files. This includes no access to the MDB parts of Manifold MFD and MapInfo TAB imports. The workaround for importing or exporting such files is to launch Manifold in 32-bit mode by using the Manifold System (32-bit) shortcut, perform the export from or import into a .map project file and then re-launch Manifold in 64-bit mode using the Manifold System (64-bit) shortcut. Linking is more complex: the data must be in some format usable in 64-bit mode within 64-bit Windows systems, such as a SQL Server database. Alternatively, the data can be kept within a Manifold .map project file and linked using the Manifold ODBC driver.

See Also

Script Examples for examples of scripting.
Other Examples for a partial list of other examples within other topics.
My First Map

This example creates a simple map of the San Francisco Bay region. It uses pre-existing files in MapInfo .mid/.mif format. These are provided as examples on the Manifold CD in the Examples/Help directory. Maps in MapInfo .mid/.mif format have two files per map: a .mif file and a .mid file.

It may be odd to begin an example of Manifold System usage by importing files from a different GIS system’s format, but one of the big strengths of Manifold is the ability to import data from many different formats. We will often import files from other formats for our Manifold projects.

For this example, we will use the files shown above as seen in Windows Explorer. Note that it is wise when using Windows Explorer in GIS usage to have the Windows Explorer View menu set to showing details, instead of using large cartoon icons, and to configure Explorer’s Tools - Folder Options - View options so that files are shown with three letter extensions.

If Explorer doesn’t know by default what a particular file is supposed to be it won’t show a sensible icon, and Explorer as set up by default knows nothing about common file formats in the GIS world. We can see that effect in the above illustration, where Explorer thinks that a .mid file is not part of a MapInfo .mid/.mif ensemble but rather represents a MIDI sequence file for use with Windows Media player.

Step 1: Launch Manifold

Manifold launches by default with the project pane open. The project pane shows all components (drawings, images, etc.) in the project. If the project pane is not open, we can open it with a View - Panes - Project command or with the SHIFT-ALT-P keyboard shortcut.

If the project pane is docked and we would like to undock it, we can hold down the CTRL key and then drag the pane away from its docked position. The CTRL key prevents it from docking again even if we drag it just a little bit away from the main window’s border. See the Windows topic for examples of docking and undocking.

Step 2: Import the bay_land_eg Drawing

Choose File - Import - Drawing to import a drawing.
In the *Import* file opening dialog (a standard Windows "Open" dialog), choose MIF Files (*.mif) in the Files of type box. Even though .mid/.mif files occur in pairs of files, one of which is a .mif file and one of which is a .mid file, the import dialog shows only the .mif files for clarity.

Browse over to the directory containing the sample files in the usual Windows way and double-click on the *bay_land_eg.mif* file to open it. We can also click on the file and push the *Open* button. If we wish to import more than one drawing in this dialog, we can CTRL-click on more than one file to highlight several and then press Open to import several files at once. We can also click on one file and then SHIFT-click on another to highlight all the files in between if we wish to choose several files for import.

The *Import MIF File* dialog opens to allow us to check which fields we wish to import. The *bay_land_eg.mif* file has no data attribute fields to import, but the dialog still opens. This lets us know there are no fields to import. Manifold can also import formatting from MapInfo files so this box is checked by default. The sample files on the Manifold CD have no elaborate formatting so it doesn’t matter whether we check this or not, but it is usually a good idea to leave this box checked. Click *OK*.

The import process creates a drawing component in the project pane. By default, the imported drawing is named after the file imported, but with the first letter capitalized. If we like we can change the name by right clicking on the component and choosing *Rename*. We can also click on the component, pause, and then click on it a second time to rename it in the usual Windows way.

**Step 3: Import other Drawings**

Repeat **Step 2** above to import two additional drawings: *bay_hydro_eg.mif* and *bay_roads_eg.mif*.

As these drawings are imported they appear in the project pane. These drawings show major hydrographic features and major roads.
Note that the project pane's toolbar icons are disabled until we highlight a component in the project window by clicking on it. Manifold tries to maintain a "quiet cockpit" where controls are not enabled for use unless they can be used.

If we move the mouse cursor over an item in the project pane Manifold will tell us about that component in a tooltip. The Bay_land_eg Drawing component, for example, is a local component: it is resident in this project and is not linked into the component from some external data source.

**Step 4: Create a Map**

To create a map using this drawing we right click on a component and choose Create - Map. In the Create Map dialog that appears, all of the components in our project that can be used to create a map will appear in a list. If we right click on the Bay_land_eg Drawing component and choose Create - Map that will launch the Create Map dialog with that component already selected to participate in the map.

![Create Map dialog](image)

We can check the boxes of each component we wish to appear as a layer in the map. For now, we will leave only the Bay_land_eg Drawing component checked. We can also change the name of the map to Bay Area in the Name box and then press OK.

![Project pane](image)

A map component called Bay Area appears in the project pane. To open it in a map window, we can double-click it.
Maps are opened in a map window, which has layer tabs at the bottom of the window. The illustration above shows a map window that has been resized into a smaller size to fit into this Help file conveniently.

**Step 5: Drag and Drop Drawings into the Map**

We can add additional drawings to the map by dragging and dropping components from the project pane into the map.
Add the **Bay_hydro_eg Drawing** component by clicking on it in the project pane and dragging it into the map window and then releasing the mouse button. It will appear as a layer in the map.

Click and drag the **Bay_roads_eg Drawing** component into the map as well.

We could have created the map with all three drawings as layers by checking their boxes during the creation of the map. Had we done so the map would have been created right away with three layers. Instead, we created the map with one layer and then dragged and dropped the other drawings into the map to show that it is possible to add layers to a map by dragging and dropping components into it.

A good reason to create a map using one component and then drag and drop other components into the map window is to guarantee that the map will use a specific projection. When a map is created it takes its initial projection from the projection of one of the components used to create the map. If all of the drawings used in a map use the same projection then no matter which of them is used for the "seed" projection for the map the projection used will be the same. However, in case drawings use different projections it is wise for the new user to create maps with one component and to then drag and drop other components into those maps. This guarantees the map will use the projection of the initial component.

After we feel comfortable with projection concepts and the use of projections in drawings and maps we can, of course, create maps consisting of many components at once and still retain perfect control over which projection is used by the map.

**Step 6: Format Drawings**

Let's change the appearance of the layers in our map. We will begin by eliminating the large round dots that show the position of points.

The **hydro** and **roads** drawings came from USGS DLG maps in which every line is terminated with a point. This is important in certain types of network analyses. Manifold displays points by default as a size 3 point since otherwise they are very difficult to see as single pixel points. When using points in drawings that show, for example, the positions of cities as a point it is important to be able to see the points. In the case of roads or other drawings where points are present for network analytic reasons we don't want to see the points so we reduce their size.
Click on the points size value and change it to 1/20.

Click on the **roads** layer (we'll abbreviate names) and in the Format Toolbar change the points size value to 1/20 as well.
That's better! The above procedure reduces points to a single pixel in size on screen (and only 1/20th of a point when printed), which makes them less visible but still leaves them in the drawings. As a technique, it is something of an expedient hack.

If we don't intend to use the roads or hydro layer for analytic purposes we might choose to delete the points. This is easy. First, make the points size 3 again so you can see them and thus be impressed by what follows. Next, we setup the selection modes (see the Selection in Drawings topic) to select points only and then we use Edit - Select by Type (or CTRL-T, the keyboard abbreviation for Select by Type) to select all points in the layer. Finally, we press Delete. Good-bye points!

To change the land areas from the default gray colors we can click on the land layer and change the color of the areas to whatever color is desired.

Click on the background color well for areas in the format toolbar and choose a new color in the resultant menu of color samples.
The color selected will be applied throughout the layer for background color for areas.

Experiment with different formats for the three layers in the map. Try different foreground and background colors for areas, and different area styles. Try changing the size or color of lines in the hydro or roads layers.

For example, to change the colors of lines in the hydro layer we first click on the hydro layer tab to make it the active layer.

We then click on the foreground color for lines in the Format Toolbar and select a blue color.
Note that since there are no lines or points in the land drawing there will be no changes visible in that layer when changing line or point formats. Likewise, since there are no areas in either the hydro or roads layer there will be no changes visible in those layers as a result of changing area format.

**Step 7: Add Labels**

To add labels to the map we first create a labels layer in the map and then we add labels as desired.

Right click onto the leftmost layer tab in the map window...

...and choose Add - New Labels from the context menu.
In the **Create Labels** dialog press the **OK** button without choosing any options. This creates a new, blank labels component in the project and adds it as a layer in the map.

![Create Labels dialog]

Click on the **Insert Label** tool...
...and then click into the map at a location where a label is to appear.

Enter the text for the label, such as Angel Island and press OK. (Since label text can consist of multiple lines, pressing Enter will start a new line. Use OK to apply the label.)
Add labels as desired. They may then be formatted using the **Format Toolbar** to change style, foreground and background color, font, size and alignment.

All of the labels in a given labels layer will use the same formatting. To add labels using different styles or colors, add another labels component and create additional labels in that component. The new component may then be formatted using different format settings.

**Step 8: Save the Project**
Examples

It's easy to save a project. Choose File - Save and browse over to the folder where the project is to be saved. Enter Bay Area as the file name and push Save. The ".map" extension will be appended to the file name given and the project will be saved as Bay Area.map

**Step 9: Print the Map**

The simplest way to print a map is to open it and then use File - Print to print it. This is a quick print but has limited control over how printing occurs. A more customizable way to print is to use the map to create a print Layout and to then print the layout.

**Step 10: Experiments**

We can perform the simple steps above in different ways to experiment with the user interface. Once the project is saved, we can do whatever we like and then reload the project. Try the following:

- Use File - Create - Map to create additional maps. Try creating them with all layers at once by checking the boxes. Try using the Move Up and Move Down buttons in the Create Map dialog to specify the order of layers.
- Open more than one window for the same map by using the Open in New Window button in the project pane's toolbar. Try panning and zooming the different windows differently.
- What happens when you delete a drawing in the project pane when it is used as a layer in an opened map window?
- Try closing the project and then opening the saved version using File - Open.
- Try using copy and paste within the project pane. For now, just paste as a drawing.
- Try importing the mexico_eg sample drawing. Open the drawing's table and click and right click on various parts of the table (column heads, cells, record handles).
- Double-click open the Bay_land_eg Drawing and also double-click open the table for this drawing. Try selecting items in the drawing window. What happens in the table window?

**Important: Verifying Projections**

This example imports files from MapInfo .mid/.mif format, which does a reasonably good job of saving projection information. In this example we did not have to worry about verifying the projections used by the imported components.

The situation is very different when importing drawings or other components from formats that do not save projection information, such as AutoCAD DXF or (in many cases) ESRI "shapefiles." In that case it is possible that importing the drawing will not result in the correct projection being assigned. A good rule of thumb as to whether or not a new component is at risk of having been imported with an incorrect projection assigned is if the new component has been imported into Orthographic projection.

If there is no projection information in a file format from which the component is imported, Manifold by default will assign Orthographic projection to that component. For example, if we import an image from a non-GIS graphics format such as ordinary .jpg, .gif, .bmp or similar the image will be assigned Orthographic projection. Therefore, if a new component uses Orthographic projection Manifold is alerted to the possibility that the component has been imported from a file format that does not provide projection information.

Because it is so important that the correct projection be assigned to a new component, whenever a new component uses Orthographic projection Manifold is alerted to the possibility that the component has been imported from a file format that does not provide projection information.

**Note:** The rule of thumb used by Manifold to decide if a component has been imported from a format that does not store projection information is to simply raise the info bar whenever a new component's projection is Orthographic, the default used for imports from formats that do not provide coordinate information, or if the component has been imported in Latitude / Longitude with coordinate locations outside the expected range (+/- 90 latitude and +/- 180 longitude). Either case is a tip-off that a component has been imported from a format that does not store projections so it makes sense for Manifold to ask us to verify the projection used.
Let's see how this would have worked if our initial drawing had been imported from a geographically unaware format and so the drawing initially had been imported into **Orthographic** projection.

When first opened in a window (in this case, in a map window) Manifold will display an info bar as seen above to ask us to verify the projection of the new component.
Clicking on the information bar at the top of the window launches the Assign Projection dialog. We can use this dialog to change the projection to be used to **Latitude / Longitude** projection and then press the **OK** button.
The map window would thereafter appear with no information bar warning about the need to verify the projection of the new component. Once a component has been verified it need not be re-verified.

Verifying the projection of a new component is a routine matter in Manifold when importing from geographically-unaware formats. To avoid the constant repetition of telling users to verify the projection when an info bar like that shown above appears, this documentation will not mention this routine step except in cases where we must go beyond mere verification to specifying the correct projection.

**Notes**

This example shows a fairly typical session creating a new map. We often create a new map by importing various layers, creating a map from one of those layers and then adding other layers to that map.

There are two artificial aspects of this example. The first is that the drawings imported had no data attributes. Drawings usually do have attributes in their tables, and we will often use those attributes to help us format different parts of the drawing.

The second aspect is more subtle and perhaps a bit unfair to beginners. It is usually wise to open a new drawing in its own drawing just after import, so that the projection can be verified, before we use that component to create a map.

Suppose we import a new drawing from some legacy format, such as ESRI shapefiles, and it turns out that we didn't know a projection had to be assigned manually. If the correct projection was not assigned and we created a map from that component anyway, well, in that case we would have just created a map using an undesired projection. It's true that we would still be invited to verify the projection and we might well catch the need to assign the correct projection to the drawing and do so.

But if we already have the map created and then alter the assigned projection of the drawing used to create the map we'll end up with a drawing in one projection and a map using a different projection, such as **Orthographic**. That is not as efficient as the case where the map uses the same projection as the drawing from which it was created and it can be confusing to boot.

For that reason after importing a component it is best to pop it open to verify the projection and only after the projection has been verified should we use it to create a map. In contrast, the procedure used in this first example was overly trusting in that we created the map first and then treated the verification of the projection of...
the drawing as a mere formality. That works in the case of a reliable format like MapInfo .mid/.mif but it doesn't work all the time if devious formats like shapefiles are being used.
**Add Points with Instant Data**

Adding points to a map is a common task. We may wish to show the locations of stores, sampling equipment, fire hydrants or other items by marking their locations with points in a map. At times we will add points by importing geocoded tables. However, Manifold also makes it easy to interactively add points to a map. We can even add data at the same time we add a point to a map.

This example shows how to interactively add points to a map, using Windows *classic* style for the illustrations. We also use *Instant Data* mode and the *Object Fields* dialog to rapidly add a name for each point. We show how to move points by editing their coordinates in a table. Finally, we add labels.

**Step 1: Prepare the map to be used.**

In this example we will use the *Bay Area* map created in the My First Map example.

Open the *Bay Area.map* project file with *File - Open*. It appears as seen above when the map is opened.
Turn off the hydro layer by double clicking its layer tab and then turn off the roads layer by double clicking its layer tab. We don't need these layers for this example.

We do need an extra drawing layer in which we will add points.

To add a drawing layer, we right click on the roads layer and choose Add - New Drawing. A new drawing will be created in the project and will be added as a layer to the map just above the roads layer.

In the project pane we rename the new drawing to Stations and the drawing's table to Stations Table. We will pretend we are adding points that represent rapid response service stations for computer hardware repair.

**Step 2: Add a field and prepare to add points**

If we double click open the Stations Table we see it has no fields except the default ID field. We will add a text field called Name to the table. This will allow us to automatically add the name of each point as we enter it.
To add a field to the table we right click on a column head (like the ID column head shown above) and choose Add - Column.

In the Add Column dialog we specify "Name" for the name of the new field and choose its type to be ANSI text. Note that fields are created as integer numeric fields by default. Change the Type to ANSI text as seen above. Press OK and the field is created.

The table is still empty since there are no objects in the Stations drawing. Before adding points we will turn on Instant Data.

To do this, click on the map window so it has the focus with the Stations layer being active and choose Edit - Instant Data to turn on the check mark. When Instant Data is on, adding any objects to the drawing will pop open the Instant Data dialog that allows us to rapidly add a value to a field. Instant Data is extremely efficient when we need to add many objects with a note placed in a data field for each. See the Instant Data topic for details.

One last preparatory step: We click into the Format toolbar and choose bright green for the background color for points and 4 for their size. This will make the newly created points clearly visible.

**Step 3: Add points using Instant Data**

Click on the Insert Point tool to begin adding points.
Position the **Insert Point** cursor where the new point is to appear and click.

When we click to insert the point the **Instant Data** setting will cause the **Object Fields** dialog to automatically pop open. It will be pre-loaded with the **Name** field ready for editing since that is the only editable field in our table.
We can add the text value *Marin* and click **OK** (or simply press **Enter**).

A point appears in the **Stations** drawing in the map where we clicked.

If we look at the **Stations Table** we can see a record has been added for the point with the text value **Marin** in the **Name** field.
We can position the **Insert Point** cursor once more in the map and click.

When **Instant Data** is on, every time we insert a point the **Object Fields** dialog will launch to allow us to add a value. The dialog will launch with the last-edited field ready for editing with the last-used value already entered.
For this point we will use the text value **San Francisco** in the **Name** field and then press **OK** (or press **Enter**).

When we press **OK** another point appears in the **Stations** layer.

In the **Stations Table** another record appears for the new point with the text value **San Francisco** in the name field.
We can continue to click into the map in this fashion to add more points. With each point added we can specify the value for the Name field for that point in the Instant Data dialog. This is very fast.

For each point the Stations Table will be updated with the values we entered.

**Step 4: Adjust positions using intrinsic fields**

After adding a point to a drawing we may wish to adjust the latitude and longitude coordinates of the point to move it to a precise position. Suppose, for example, we've visited Intel in Santa Clara and have noted the exact coordinates of our service station location with a GPS. We would like to move the point we added for Intel to the exact latitude and longitude measured with the GPS.

We can accomplish this by showing intrinsic fields in the table and changing the values of those fields. Intrinsic columns are not shown by default in tables. We can use the View - Columns command to show intrinsic fields. See the Intrinsic Fields in Tables topic for details on intrinsic fields.
Turn on intrinsic fields by clicking open the Stations Table table and then choosing View - Columns. In the Columns dialog check the boxes next to the Longitude (I) and Latitude (I) intrinsic fields and press OK. The intrinsic fields will appear in the table. To change the Longitude (I) position for Intel we simply double click into the cell, specify the value desired and press Enter.

The intrinsic fields will appear in the table. To change the Latitude (I) value we likewise double click into the cell and edit the value as desired. The point will move to the new location specified by the intrinsic fields. [Depending on refresh settings, the points may not actually move in the drawing unless the drawing window is closed and then opened again.] Note: these values are made up values for the sake of this example. Intel is not located exactly at the position shown.

**Step 5: Add labels to the map**

If we like, we can use the Name field we’ve added to create labels for the new points. To do this, right click on the Stations drawing component in the project pane and choose Create - Labels in the context menu.
The Create Labels dialog will appear. The dialog knows we right clicked on the Stations drawing so it will highlight that drawing in the upper, Parent, pane assuming we want to create labels using the fields in that drawing.

We double click on the Name field in the Columns pane to add it as part of the label text in the lower pane. The other fields listed are intrinsic fields that we are not interested in using. Press OK to finish. See the Creating Labels from Fields topic for details on creating labels using this dialog.

The result is a new labels component called Labels that appears in the project pane underneath the Stations drawing. To show the labels in the map we drag and drop the Labels labels component from the project pane into the map window.
We can specify the alignment of the labels using the alignment controls in the format toolbar for labels. In the illustration above we’ve specified that the labels should appear below and to the left of their tie points. See the Aligning Labels topic for details.

**Latitude and Longitude Notation in Editing**

Manifold tables show latitude and longitude intrinsic fields using decimal degrees notation by default. When editing intrinsic fields, Manifold allows entry of values in either decimal or in degrees, minutes, seconds notation using almost any separator between the degrees, minutes and seconds values.

For example, if we want to enter a value of -122.55125 degrees we can double click into the Longitude (I) box and enter any of the following:

- -122.55125
- -122d33m4.5s
- -122 33 4.5
- -122^33'4.5"
- 122.55125W
- W122d33m4.5s

Manifold will interpret any non-numeric character in between the numbers other than a decimal point as a separator between degrees, minutes and seconds values. The letters W and S either before or after the other characters are commands to make the value a negative (west) longitude or negative (south) latitude.

Note that case is significant in NESW notation, so that 22d33m4.5s would be a North (positive) latitude and 22d33m4.5S or 22d33m4.5sS would both be South (negative) latitudes.

Whatever notation style is used to enter the new value, it will be displayed using the column format specified for that column.
Examples

Create a Projected US Map

Projections are used to give geographic maps a more natural appearance. They also allow accurate measurement in many ways not otherwise possible. We will often acquire GIS data in unprojected ("Latitude / Longitude Projection") form. This example shows how to present such data in projected form.

Important: If you have installed 64-bit Manifold, do this example in 32-bit mode by launching Manifold System using the Manifold System (32-bit) shortcut. See the note on 64-bit Manifold at the end of this topic.

We begin with an unprojected drawing of the United States. To show this in projected form we have two choices:

- Project the drawing into the desired projection and then use it to create a map. The map will automatically use the projection used by the drawing originally used to create the map. When creating maps from large drawings or images this results in faster display.
- Leave the drawing in unprojected form and then use it to create a map. We can then change the Edit - Assign Projection setting in the map to use whatever projection is required. This method is flexible because it leaves the drawing in the original state in which it was imported. However, with larger drawings or images it will result in slower display speed for maps because the native data within each drawing or image in the map must be re-projected on the fly to display in the desired map projection.

In the first part of this example we import a drawing of the United States and then project it into Lambert Conformal Conic projection. We then create a map from the projected drawing. In the second part of this example we import the drawing and leave it in unprojected Latitude / Longitude projection. We use it to create a map and then we change the map's projection.

This example uses a file in .mfd/.mdb format. In this format, there are two files, one ending in .mfd and one ending in .mdb, both with the same name, that make up the GIS drawing. When copying from the Manifold CD to your hard disk, make sure to copy both files. For example, when copying the US_Main drawing from the Manifold CD to your hard disk (if you choose to do so), don’t forget to copy both US_Main.mfd and US_Main.mdb together to the same hard disk folder.

First Method: Re-project the drawing and create a map.

Step 1: Import and open a drawing

We will use the US_Main.mfd example drawing. Launch Manifold and choose File - Import - Drawing with the Files of type box set to MFD Files.

| File name: | US_Main.mfd | Open |
| Files of type: | MFD Files (*.mfd) | Cancel |

Browse over to where the US_Main.mfd file is kept on your system and click on it to load the File name box in the Import dialog as seen above. Press OK to import the file. Press OK again in the Import MFD File dialog (not illustrated) to import the file with all fields by default.

The result in the project pane is a new drawing and its associated table. Double click the drawing name to open it in a drawing window. (A tool tip pops up to let us know this drawing is a local drawing, that is, it is stored entirely within the project file as opposed to being linked in from some remote data source.)
The drawing window shows us the **US_Main Drawing** as seen in the "unprojected" *Latitude / Longitude* projection.

**Step 2: Re-project the drawing**

Choose Edit - Assign Projection to confirm that the projection assigned to the drawing on import is correct. The command will launch the **Projection** dialog to **confirm** the projection assigned. Click OK. Now, choose Edit - Change Projection to project the drawing to another projection. The command will launch the **Projection** dialog to **change** the projection used by this drawing.

**Note:** We use Assign Projection to verify the projection before Change Projection because Manifold wants us to verify the projection of a drawing is correct before it will allow us to change the projection. Trying to re-project a drawing where the original projection was not correctly assigned will cause chaos, so Manifold insists we check that initial assignment of projection is accurate.

The **Latitude / Longitude** projection is highlighted in the projections tree to show us that is the projection currently in use for this drawing. We click on the **Conic** hierarchy to expand it.
In the expanded Conic projections hierarchy we choose Lambert Conformal Conic. This projection is well suited for maps of the lower 48 US states. See the Projections topic for discussion and links to other topics that provide advice on choosing projections.

Clicking on Lambert Conformal Conic loads the parameters pane with the parameters (such as the Center Latitude and Center Longitude) that may be set for this projection. By default the projection parameters are all set to 0.00. To provide sensible parameters for this drawing we press the Suggest button.
The Suggest button loads the projection parameters with suggested values to use. Note that latitude 37 and longitude -96 is about the center of the United States. See the Lambert Conformal Conic topic for information on suggested settings for these parameters. We press OK to re-project this drawing.

The result is that the drawing is re-projected into Lambert Conformal Conic projection using the parameters specified.

**Step 3: Create a map from the drawing**

If this single drawing is all we will use, there is no need to create a map. We can perform formatting and print directly from the drawing window if so desired. However, if we intend to have a map with other layers we can create a map from the drawing. We can then add other layers to the map.
To create a map we click on the **Create** button in the project pane.

In the pull-down menu that appears we choose **Map** to launch the **Create Map** dialog.

The **Create Map** dialog lists only one component, the **US_Main Drawing**, because it is the only component in our project so far. If we had other components in the project these would be listed as well. We change the name to **US Map**. By default the checkbox is checked for this component so all we need do is press **OK** to create the map.

A new map is created in the project. To open it we double click on the map name in the project.
The result is a map that shows our drawing. Because the map was created using a drawing that was in Lambert Conformal Conic projection, the map also uses this projection for its view.

**Second Method: Create a map from the drawing and change the map's projection**

In the method used above we imported a drawing, re-projected the drawing and then created a map. We could achieve the same visual effect in the map by importing the drawing, creating a map and then changing the projection used to show the view in the map.

The distinction is that in the first case the drawing is modified to a new projection so that when the map shows this drawing in that same projection both the drawing and the map are using the same projection. In the second case, the drawing is not modified: it retains the original projection. The map, however, will dynamically re-project the data within the drawing to show it in whatever projection view we choose.

The first step using the second method is to import the drawing. However, since we will not re-project the drawing there is no need to open it.

**Step 1: Import a drawing**

We will again use the US_Main.mfd example drawing. Launch Manifold and choose File - Import - Drawing with the Files of type box set to MFD Files.

Browse over to where the US_Main.mfd file is kept on your system and click on it to load the File name box in the Import dialog as seen above. Press OK to import the file. Press OK again in the Import MFD File dialog (not illustrated) to import the file with all fields by default. The result in the project pane is a new drawing and its associated table.

**Step 2: Create a map from the drawing**

Even if this single drawing is all we will use in our project in this case we will need to create a map to see the drawing in the desired projection.
To create a map we click on the Create button in the project pane.

In the pull-down menu that appears we choose Map to launch the Create Map dialog.

As before, the Create Map dialog lists only one component, the US_Main Drawing, because it is the only component in our project so far. We change the name to US Map and press OK to create the map.
A new map is created in the project. To open it we double click on the map name in the project.

The result is a map that shows our drawing. Because the map was created using a drawing that was in Latitude / Longitude projection, the map also uses this projection for its view.

**Step 3: Change projection used by the map**

Changing the projection used by a map to show its contents is very similar to re-projecting a component except that the **Edit - Assign Projection** dialog is used instead of the **Change Projection** dialog. The **Assign Projection** dialog is used because altering the projection used by a map does not re-compute the coordinate data inside the components used in the map: it simply changes the projection used by the map to view the layers in the map. To change the projection used by this map we choose **Edit - Assign Projection**.

The Latitude / Longitude projection is highlighted in the projections tree to show us that is the projection currently in use for this map. We click on the **Conic** hierarchy to expand it.
In the expanded **Conic** projections hierarchy we choose Lambert Conformal Conic.

To provide sensible parameters for this map we press the **Suggest** button.
The **Suggest** button loads the projection parameters with suggested values to use. It will suggest the same values as seen in the first method since the drawing in use in the map is the same. Press **OK** to change the projection view used by the map.

The result is that the map now shows its contents in the **Lambert Conformal Conic** projection using the parameters specified. The new projection is computed on the fly without changing the contents of the **US_Main Drawing** component.
In fact, if we double-click open the **US_Main Drawing** component we can see that it is still in **Latitude / Longitude** projection.

Leaving the **US_Main Drawing** in its original **Latitude / Longitude** projection is not a bad idea if this drawing will participate in other maps in our project that will use different projections. It takes computational overhead to re-project a drawing on the fly to display within a map that uses a different projection, but if the data set is small (that is, few coordinates defining the objects in the map and few objects in the map), then the overhead is not too burdensome. If we have large data sets (drawings with many, very detailed objects) then it pays to re-project the drawing or the map so that both use the same projection.

This example drawing is a relatively small data set so we can easily use it within different maps that use different projections and it will be re-projected on the fly for display.

**Using US_Main in a world map**

Suppose for example we wish to use **US_Main** as a layer in a world map. Using a procedure analogous to that shown in the second method above, we will create a world map using the **World_eg.mfd** example file. To do so we will:

- Import **World_eg.mfd**.
- Use **Edit - Assign Projection** to verify the projection used by the **World_eg** drawing.
- Create a map using the **World_eg Drawing** component as the only layer and name it **World**.
- Open the **World** map.
- Use **Edit - Assign Projection** to change the projection used by the **World** map into the Robinson projection.
At the end of the above sequence the World map will appear as seen above. Note that it shows its contents using the Robinson projection even though the World_eg drawing is in Latitude / Longitude form.

To add the US_Main Drawing to the World map we click on it in the project pane and drag and drop it the highlighted drawing from the project pane into the opened World map window.
The result is that the **US_Main Drawing** appears as a layer in the **World** map. In the illustration above we have taken a moment to click on the **Background Color** well for areas in the **US_Main Drawing** and have changed it to bright yellow. This makes the **US_Main Drawing** layer more obvious in the map.

We can have multiple windows open at once that show the **US_Main Drawing** component in different projections. In the upper left of the screenshot above we see the component open in a drawing window where it is seen in its native **Latitude / Longitude** projection. Drawing windows always show their drawing in the native projection of that drawing.

In the lower left we see the **US Map** map window that has been instructed to show its contents using **Lambert Conformal Conic** projection. At right we see the **World** map window that has been instructed to show its contents in the **Robinson** projection.
Note that changing the format of a drawing changes the appearance of that drawing in all cases. When we changed the background color of areas to yellow for **US_Main Drawing** the color change was updated in all three windows.

**Note**

The screen shot above shows an actual Manifold session open with three windows. The windows have been resized to be very small so that the screen shot can be made small enough to fit into this document. In normal usage of course the windows would be larger. Manifold users often will have large screen monitors with high resolution so they can keep several conveniently large windows open at the same time.

An interesting note for Manifold users who are unfamiliar with the world of legacy GIS applications is that some old-fashioned GIS packages can’t show more than one window open at the same time. Surprisingly, even though they may cost thousands of dollars they not only cannot show the same drawing re-projected on the fly in multiple windows, they can’t even show more than one window open at a time. Amazing!

### Important Note when Using 64-bit Manifold Editions

Due to a lack of required Microsoft facilities in 64-bit Windows systems, Manifold in 64-bit mode cannot export, import, export or link to DB, HTML, MDB, XLS or WKx format files. This includes no access to the MDB parts of Manifold MFD and MapInfo TAB imports. The workaround for importing or exporting such files is to launch Manifold in 32-bit mode by using the Manifold System (32-bit) shortcut, perform the export from or import into a .map project file and then re-launch Manifold in 64-bit mode using the Manifold System (64-bit) shortcut. Linking is more complex: the data must be in some format usable in 64-bit mode within 64-bit Windows systems, such as a SQL Server database. Alternatively, the data can be kept within a Manifold .map project file and linked using the Manifold ODBC driver.

### See Also

- **Maps**
- **Edit - Assign Projection**
- **Edit - Change Projection**
- **Using Projections**
Examples

Using Layers

Layers in maps are components such as drawings, images, surfaces or labels components. We can use layers to organize our data and to present it through good formatting. This example shows how to move objects between drawing layers and how to format different layers.

**Important:** If you have installed 64-bit Manifold, do this example in 32-bit mode by launching Manifold System using the Manifold System (32-bit) shortcut. See the note on 64-bit Manifold at the end of this topic.

In this example we begin with a map with one layer, the US_Main Drawing projected map created in the Create a Projected US Map example. We then select groups of states and move them into new layers to create a map with three layers.

Recall that layers in a map can be images, drawings, surfaces or labels components. Every time we add a drawing layer to a map we create a new blank drawing in the project pane. When a map consists of drawings only, each layer in the map is a drawing.

**Step 1: Copy and paste western states to a new drawing layer**

Import US_Main.mfd and project the drawing into Lambert Conformal Conic projection. Create a map using this drawing. Use the procedures in the Create a Projected US Map example.

We see the map in the illustration above. The drawing layer has been renamed US_Main for brevity.

Set up Add to Selection selection mode, Select Areas selection objects and the Select Touch selection tool as seen above. Each area we touch with the mouse will be added to the selection.
Touch eleven western states as seen above.

Cut the states out of the US_Main layer using CTRL-X or Edit - Cut.

Right click onto the US_Main layer tab and choose Add - Drawing to create a new drawing layer. Call this drawing West.
Paste into this layer using CTRL-V or Edit - Paste.

The Paste Objects dialog appears asking how to paste the fields associated with these objects. The US_Main drawing has several fields for each state. When the West drawing layer was created it had no columns in the drawing's table for that layer. If we use the default settings in this dialog it will create new columns in the table for the West drawing layer. Press OK to use the defaults.

The eleven western states appear in the West layer. Newly pasted objects are selected.

We can deselect them by pressing the Select None button. They will then appear in the default gray color format.

To change the color of the western states, we click on the background color well for areas in the Format toolbar.
Choose a light yellow color in the color menu that appears.

Changing the color to light yellow in the toolbar will change the color of the objects.

The western states areas will now appear using an areas background color of light yellow.

**Step 2: Copy and paste central states to a new drawing layer**

Copying states from a central region to a new layer uses the same procedure as in the preceding step.
Click on the **US_Main** layer to make it the active layer. Click on the states shown to select them (the selection tool should still be set to **Select Touch**). If we do not first click on the **US_Main** layer to make it the active layer we will not be able to select these states because they would not be in the active layer. Selection and Clipboard (Cut, Copy, etc) operations in maps work on the active layer by default. To extend the action of these operations to all visible layers, press the **ALT** key.

Press **CTRL-X** or use **Edit - Cut** to cut the states out of the map.
Right click onto the US_Main layer tab and choose Add - Drawing to create a new drawing layer. The new layer tab will be created to the left of whatever tab was right-clicked. Call this new drawing Central.

A brief digression: We can alter the appearance of the map window by turning the background off and on.

If we open up the Layers pane with a SHIFT-ALT-L or with a View - Panes - Layers menu command we can uncheck the Background box to turn off the background.

Without the background, the default checkerboard background pattern shows through. Note that we can also turn layers off and on very quickly by checking and unchecking their boxes in the Layers pane.

Check the Background box again to restore the white background.
Click on the **Central** layer tab to make it the active layer. Press **CTRL-V** or **Edit - Paste** to paste the objects in the clipboard into the layer. Once again the **Paste Objects** dialog will be raised (not illustrated). Press **OK** to accept the defaults in the dialog and to paste into the layer. The central states will appear in red selection color. Press **Select None** to deselect them.

To format areas in this layer a color different than gray, click on the background color well for areas in the **Format** toolbar.

Choose a light green color from the color menu that appears.

The color well in the toolbar will show the new color selected.
All areas in the **Central** layer will be colored using the new format.

**Step 3: Rename US_Main layer**

To provide a non-default color for the remaining layer, we click on the **US_Main** layer to make it the active layer and then click on the background color for areas in the format toolbar to change the color.

We can change the color to a light blue. To complete our map dividing the US into three regions, in the project pane we rename the **US_Main** drawing to **East**.

The layer tab for this drawing in the map will also automatically be renamed.

**Step 4: Moving objects between layers**

We've completed the map we set out to create. If desired, we can make changes in it by moving objects between the layers created. For example, suppose we decide that Michigan should be assigned to the **East** layer instead of the **Central** layer.
Click on the **Central** layer tab to make it the active layer and then click on Michigan to select the state.

Press **CTRL-X** or **Edit - Cut** to cut the state out of the **Central** layer. Click on the **East** layer tab to make it the active layer.

Press **CTRL-V** or **Edit - Paste** to paste Michigan into the **East** layer. Note that the **Paste Objects** dialog is not raised because the field structure from the origin layer matches the destination layer.
If we press Select None to deselect Michigan we see that it appears in the light blue format color used for areas in the East layer.

**Important Note when Using 64-bit Manifold Editions**

Due to a lack of required Microsoft facilities in 64-bit Windows systems, Manifold in 64-bit mode cannot export, import, export or link to DB, HTML, MDB, XLS or WKx format files. This includes no access to the MDB parts of Manifold MFD and MapInfo TAB imports. The workaround for importing or exporting such files is to launch Manifold in 32-bit mode by using the Manifold System (32-bit) shortcut, perform the export from or import into a .map project file and then re-launch Manifold in 64-bit mode using the Manifold System (64-bit) shortcut.

Linking is more complex: the data must be in some format usable in 64-bit mode within 64-bit Windows systems, such as a SQL Server database. Alternatively, the data can be kept within a Manifold .map project file and linked using the Manifold ODBC driver.
**Color a Drawing**

At times we would like to color a drawing of areas so that no two adjacent areas have the same color. To do so, we use the Color dialog to automatically color areas.

**Important:** If you have installed 64-bit Manifold, do this example in 32-bit mode by launching Manifold System using the **Manifold System (32-bit)** shortcut. See the note on 64-bit Manifold at the end of this topic.

**Step 1: Import and open a drawing**

We've imported a drawing of US states (the **US_Main.mfd** sample drawing) and have projected it into Lambert Conformal Conic projection. Seen in default colors, all of the states are drawn using default light gray background color. To import and project the drawing used the procedures described in the previous Example topic, Create a Projected US Map.

We've imported a drawing of US states (the **US_Main.mfd** sample drawing) and have projected it into Lambert Conformal Conic projection. Seen in default colors, all of the states are drawn using default light gray background color. To import and project the drawing used the procedures described in the previous Example topic, Create a Projected US Map.

![US Main Table](image)

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>State_FIPS</th>
<th>Abbrev</th>
<th>FIPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Minnesota</td>
<td>27</td>
<td>MN</td>
<td>27000</td>
</tr>
<tr>
<td>2</td>
<td>Washington</td>
<td>53</td>
<td>WA</td>
<td>53000</td>
</tr>
<tr>
<td>3</td>
<td>Idaho</td>
<td>16</td>
<td>ID</td>
<td>16000</td>
</tr>
<tr>
<td>4</td>
<td>Montana</td>
<td>30</td>
<td>MT</td>
<td>30000</td>
</tr>
<tr>
<td>5</td>
<td>North Dakota</td>
<td>38</td>
<td>ND</td>
<td>38000</td>
</tr>
<tr>
<td>6</td>
<td>Michigan</td>
<td>26</td>
<td>MI</td>
<td>26000</td>
</tr>
<tr>
<td>7</td>
<td>Maine</td>
<td>23</td>
<td>ME</td>
<td>23000</td>
</tr>
<tr>
<td>8</td>
<td>Wisconsin</td>
<td>55</td>
<td>WI</td>
<td>55000</td>
</tr>
<tr>
<td>9</td>
<td>Oregon</td>
<td>41</td>
<td>OR</td>
<td>41000</td>
</tr>
</tbody>
</table>

If we open the table for the drawing we can see it has five fields.

**Step 2: Launch the Color Dialog**

Click on the opened drawing so it has the focus and choose **Drawing - Color** from the menu.

![Color dialog](image)
We use the default settings in the **Color** dialog. See the Color topic for details on operating this dialog.

The nearly instantaneous result is a drawing that's been colored so that no two adjacent states have the same color. Only four colors have been used. The drawing has been colored by creating a thematic format for the area colors.

The **Color** dialog creates a thematic format for both the **background** color as well as the **foreground** color as can be seen by the format toolbar colors for areas seen above. The foreground color theme is a darker version of the background color with the expectation that an area style using foreground color for apparent boundary is being used.

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>State_FIPS</th>
<th>Abbrev</th>
<th>FIPS</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Minnesota</td>
<td>27</td>
<td>MN</td>
<td>27000</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Washington</td>
<td>53</td>
<td>WA</td>
<td>53000</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>Idaho</td>
<td>16</td>
<td>ID</td>
<td>16000</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>Montana</td>
<td>30</td>
<td>MT</td>
<td>30000</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>North Dakota</td>
<td>38</td>
<td>ND</td>
<td>38000</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>Michigan</td>
<td>26</td>
<td>MI</td>
<td>26000</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>Maine</td>
<td>23</td>
<td>ME</td>
<td>23000</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>Wisconsin</td>
<td>55</td>
<td>WI</td>
<td>55000</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>Oregon</td>
<td>41</td>
<td>OR</td>
<td>41000</td>
<td>3</td>
</tr>
</tbody>
</table>

If we open the drawing’s table we can see that a new field called **Color** has been added to the table. This field contains the value 1, 2, 3 or 4. The field is then used in the thematic format to color the drawing.

**Step 3: Adjust Colors if Desired**

It’s easy to change the color scheme after applying the color dialog. Simply click on the background color well for areas and choose **theme** and then apply one of the preset color schemes. The color applied by the color dialog is simply a thematic format using the **Color** field with a **Unique Values** method. If we choose a different color preset we can alter the colors at will.
First we click into the foreground color well and change it to dark gray. Dark gray boundary lines will look better with whatever new color scheme we choose.

We click on the background color well for areas to change the thematic format for areas.

In the pull down color choice menu we choose Theme to launch the thematic formatting dialog.

In the thematic formatting dialog we choose Color for the field and Unique Values for the method and Yellows for the palette.

We press Apply to apply the color range and then press OK.
The result is a thematic format using a range of yellow colors. Instead of using a preset like Yellows, of course, we could have clicked into the color wells in the thematic formatting dialog and set whatever colors we liked.

**Extra Credit: Adjust Area Styles**

We can use thematic formatting with the **Color** field to adjust area styles instead of (or in addition to) color. Click into the foreground color and background color wells for areas in the format toolbar to change them to dark gray and very light gray respectively.

This eliminates the thematic format and results in the map seen above. However, the "coloring" values in the **Color** field are still there in the drawing's table.

We can use those values to do a thematic format that changes area style.
To do so, click on the styles well for areas and choose Theme.

In the Format dialog choose Color as the controlling field and Unique Values as the method. Click into the various styles wells for the values 1 through 4 and change them to a desired area style.

The result is a drawing where no two adjacent areas have the same area styles.

Important Note when Using 64-bit Manifold Editions

Due to a lack of required Microsoft facilities in 64-bit Windows systems, Manifold in 64-bit mode cannot export, import, export or link to DB, HTML, MDB, XLS or WKX format files. This includes no access to the MDB parts of Manifold MFD and MapInfo TAB imports. The workaround for importing or exporting such files is to launch Manifold in 32-bit mode by using the Manifold System (32-bit) shortcut, perform the export from or import into a .map project file and then re-launch Manifold in 64-bit mode using the Manifold System (64-bit) shortcut.

Linking is more complex: the data must be in some format usable in 64-bit mode within 64-bit Windows systems, such as a SQL Server database. Alternatively, the data can be kept within a Manifold .map project file and linked using the Manifold ODBC driver.

See Also

See the Color topic for details on using the wizard as well as notes on performance with large numbers of areas.

See the Thematic Formatting topic for general notes on the thematic formatting dialog used above.

See the Thematic Formatting Example topic for an example of thematic formatting.
Examples

Import a Drawing from the Geocoding Database

Drawings may be imported from the Manifold Geocoding Database that is provided for free download from the manifold.net website. The geocoding database provides information on streets and addresses in the United States that is organized on a county basis, but it also includes supplementary cartographic information. Although the principal use of the geocoding database is to provide address range data for street address geocoding the data may also be used to create drawings that show streets and hydrological features within the United States.

Like the rest of Manifold's street address geocoding capability, import from the geocoding database becomes operational only when the optional Manifold Geocoding Tools package is installed. If you do not have the optional Geocoding Tools package, you will not be able to take advantage of the import capability described in this topic.

See the Import Drawing - Geocoding Database topic and the Geocoding Data Sources topic for additional information, including instructions for installing the Manifold Geocoding Database onto your hard disk. If you have not installed the Manifold Geocoding Database, you will not be able to follow the procedure described in this topic.

This example shows how to import a drawing for Placer County, California. We then separate hydrological features into a second drawing and combine the two drawings into a map. Even those users outside the United States who are not interested in drawings of counties in the US should still read this example for an illustration of the technique of using a data attribute field to select objects within a drawing and move them to a separate drawing.

Step 1: Import Drawing

Choose File - Import - Drawing to launch the Import Drawing dialog.

In the Import Drawing dialog choose Geocoding Database Files in the Files of type box and then browse over to the folder in which the Manifold Geocoding Database files have been installed. Double click on the states.dat file.

Note: to use this procedure you must know where the geocoding database has been installed and be able to find the states.dat file. By default, the geocoding database is installed in the C:\Program Files\Manifold
System\GCDB folder. If you or someone else installed the geocoding database in a different location, such as on a network share, make note of where it was installed and browse to that folder.

In the resulting Import Geocoding Database dialog specify Placer, CA as the county name. Note that even though the technical name of the county may be "Placer County" we don’t use the "County" appendage.

All fields are checked for import by default and for the sake of this example we don’t bother un-checking any fields. Since most people will use this particular import capability to create background maps of streets or other features in applications where technical details of address ranges and other fields are not required, in many cases users will import only the Street and CFCC fields. But for now we just import everything and click OK.

The result is that a new drawing appears in the project pane.
If we open that drawing we see that it shows features in Placer county and has already been thematically formatted by default.
Clicking open the thematic format for foreground color of lines, we can see that the thematic format uses the CFCC field to thematically format colors based upon the various feature codes in the CFCC field. Even if we were unfamiliar with the standard values used in by the Census Bureau to encode different features in the CFCC field, we could easily guess that feature codes beginning with an H represent hydrology, that is, water, features.

See the Import Drawing - Geocoding Database topic for a list of CFCC codes.

**Step 2: Select Hydrology Features**

As imported, the drawing contains all features available within the geocoding database. Features have been thematically formatted to set them apart. That is a typical way to use a drawing that contains many features, especially if it is just intended as a background for other layers.

At times it may be more convenient to separate out the contents of a drawing into different drawings, for example, to separate out hydrology features into a separate drawing. Or, perhaps we would like to simplify a drawing by removing features that are not of interest. Doing so within Manifold is easy: we select the desired features and Cut them out of a drawing. We can then Paste them into a new drawing or simply discard them. Let's use hydrology features as an example.

To select all hydrology features we use the selection toolbar to select all CFCC features that contain the letter H.

Since CFCC codes only contain one letter, the initial letter, using a relatively simple selection method such as "containing H" will select all records for hydrology features since all of the CFCC codes for such features begin with H, as we can see by reviewing the table of CFCC codes given in the Import Drawing - Geocoding Database topic.
When we press the **Select** button in the toolbar all of the hydrology lines (those lines with an *H* in their CFCC field) are selected.

**Step 3: Cut Hydrology Features and Paste as New Drawing**

To **Cut** the lines out of the **Placer** drawing we click on the drawing to make sure it has the focus and then we do either an **Edit - Cut** or simply a **CTRL-X**, the standard Windows keyboard shortcut for Cut. Doing a Cut as opposed to a **Delete** will not only delete the lines from the drawing but will also **Copy** them to the Windows clipboard.
The lines disappear from the Placer drawing. To put them into another drawing, we can right click onto the project pane and choose Paste to paste them from the Windows clipboard into a new drawing. We could also use, if desired, the Windows keyboard shortcut for Paste by doing a CTRL-V.

The result is that a new drawing, called Placer Drawing 2 appears in the project pane.
If we open that new drawing we see it contains the hydrology lines that were cut out of the original Placer Drawing.

**Step 4: Create a Map**

It’s often the case that we would like to keep different classes of lines, such as streets or hydrology lines, within different drawings and then show those different drawings together as layers in a map. Creating a map is easy.

We begin by renaming Placer Drawing 2 into a more descriptive name. We’ll call it Placer Hydro. Next, we right-click on the Placer Hydro component in the project pane and choose Create - Map.

In the Create - Map dialog we check the boxes for both Placer Drawing and Placer Hydro and we press OK.
When we open the resulting map we see it contains both drawings as layers.

**Notes**

Placer County gets its name from the nomenclature of the California Gold Rush in the late 1840’s. A **placer** is a deposit of gold-rich sand or gravel, as are found in the Western foothills of the Sierra Nevada mountain range on the border between California and Nevada in the United States.

Placers were formed tens or hundreds of thousands of years ago when streams eroded the mountains and carried bits of gold worn from the hard rock of the Sierras downstream. Where streams took a bend, the rushing water slowed down and the reduced velocity of flow allowed the very heavy gold to settle at the bend, often at the same location where a gravel bank or deposit of sand would also build up.

Over time the streams dried up and their beds and banks were covered up by soil and scrubland and forests, but the beds where gold accumulated remained. A lucky or skillful miner might find such a placer and discover extremely rich concentrations of gold. Placer County stretches from the California central valley Eastward up into the foothills and mountains of the Sierra Nevada. It includes territory in which some of the richest gold strikes in history were made.

Although made famous by the California Gold Rush, placer deposits occur throughout the world, wherever erosion of gold-bearing rock and the flow of ancient streams have accumulated a mix of gold and sand or gravel.

**See Also**

Import Drawing - Geocoding Database
Fun with Google Earth

If we have drawings or images we would like to publish in Google Earth we can export them to KML or KMZ files. Double-clicking a KML file will automatically open it in Google Earth (assuming Google Earth has been installed on the computer system), so using KML is a great way of sending GIS data to our friends when the intent is mainly visual.

We can send a KML file as an attachment to email or provide it on a web site. Our friends can click it open and then see what we want them to see in Google Earth. Although Google Earth has some limitations, such as an inability to work with more than two simple data fields, that make it unsuited for sophisticated GIS data, KML files allow us to use Google Earth as a free “GIS viewer” for simple data.

A KMZ file is simply a “zipped” KML file. KMZ is smaller because it is compressed but it is not immediately human-readable unless it is unzipped. KMZ is preferred for exporting images to Google Earth because the image will be embedded within the KMZ file.

Projections and KML Format

Google KML and KMZ formats support Latitude / Longitude projection using the WGS 84 Auto datum only. Drawings to be exported to KML or KMZ can be in any projection and will be re-projected on the fly into Latitude / Longitude during export; however, to avoid a potentially slow re-projection process it is wise to explicitly re-project drawings into Latitude / Longitude using the WGS 84 Auto datum before export. This not only avoids a slow export, it also provides an opportunity to catch errors before export if the re-projection results in a strange display as might happen if the initial projection was not correctly assigned, say, after importing projected data from some format that does not store projection information.

Unlike drawings, images must be in Latitude / Longitude projection to be exported to KML or KMZ.

Exporting Drawings with Points to KML

It’s easy to export drawings with points to KML.

Consider the table showing geocoded addresses of sushi restaurants near Menlo Park that was seen in the Linked Drawings from Geocoded Tables topic as well in the various street address geocoding topics.

In the Manifold project pane, Copy the table and then Paste As a drawing. The default dialog for pasting will know the table being pasted is Latitude / Longitude data and will pre-load column choices for us.
We rename the drawing to *Sushi Spots* and open it as seen above. We’ve formatted the points to a brighter color and have changed the point style a bit.

With the drawing open we choose *File - Export - Drawing* and then use *Sushi Spots.kml* as the file name. In the *Save as type* box we choose *KML / KMZ Files* and press *Save*. In the *Export KML File* dialog that pops up we choose the *Name* field and the *Address* field as the description. Press *OK*. That’s all!

We can now send the resulting *Sushi Spots.kml* to our friends, post it on web sites or whatever. Double-clicking the *Sushi Spots.kml* open launches Google Earth and causes it to zoom down from space to the Menlo Park / Palo Alto area where we can see our locations as above.
With only two fields to work with, Google Earth does not give us much in the way of data manipulation or display capability, but if we use it wisely we can still get useful effects. For example, using the name of the restaurant for the **Name** field works just fine as a label, and using the **Address** field (just the street address part of it) works well as a description in the auxiliary Google Earth information pane.

The above technique is simple and fast, especially if we remember to use drawings that are in **Latitude / Longitude**. The technique is nearly impossible to get wrong.

This example uses points that came out of a table, but we could have used any drawing of points. For example, we could have been using a drawing that showed fire hydrant locations collected by the public works staff for our city, or we could be doing surveys of parks and noting individual trees by GPS, citing abandoned mine locations, navigation buoys, dealer locations or anything else of interest.

**Exporting Drawings with Lines to KML**

Let's dial up the fun some more by exporting a drawing that contains lines to Google Earth. We will use the drawing of Placer County, California, created in the previous Import a Drawing from the Geocoding Database example.

The drawing appears above using new thematic formatting. Because Google Earth has a dark photographic image as background it is best to use very bright colors for lines so they display well in Google Earth. We created the above formatting by clicking on the format toolbar color well for line foreground color.
We then changed the thematic format used by default when importing from the geocoding database to brighter colors, as seen above.

Exporting to KML is trivial. With the drawing open we choose File - Export - Drawing, in the Save as type box we choose KML / KMZ Files, provide a suitable file name such as Placer County.kml (don't forget the .kml extension) and press Save.
Examples

Double-click open the .kml file and Google Earth launches and displays the contents on a background of local imagery, as seen above.

Zooming in a bit we see that even in distant areas and despite the intrinsic imprecision of Google Earth the correlation between government GIS data as used in the geocoding database and Google Earth is excellent. Google Earth adds some roads and roads labels by default, which is often useful and improves the display.

We can do a lot of work in Google Earth just using lines and points. Lines are especially popular to show tracks downloaded from GPS devices (see the Working with GPS Receivers topic) or for displaying any drawings of lines, as are created so often in ordinary GIS work. The next time you take your bike off-road for a cruise through the Sierra Nevada, leave a GPS device running on the bike to build a track. When you get home, download the track into a Manifold drawing, export it to KML and send the KML file to your friends so they can see where you’ve been on Google Earth.

Exporting Drawings with Areas to KML

Google Earth can also show areas, and Manifold can export areas to KML. The procedure is the same as exporting points or lines. Drawings exported to KML may include a mix of points, lines and areas. Points and lines are exported to KML using solid, opaque color, while areas are exported using partial transparency.
Consider a drawing showing selected congressional districts from the 109th Congress. The districts have been selected as exceptionally Gerrymandered based on their perimeter to area ratio. Districts are shown as areas, colored to be visually distinct, with black color for area border lines.

"Gerrymandering" is a method of rigging elections used in the United States by both parties. The practice nullifies the votes of the targeted minority party by drawing congressional districts so that the voters are grouped into voting districts that the minority party cannot win. It is a highly anti-democratic and corrupt practice that, of course, is enthusiastically embraced by incumbent politicians.

The drawing above is shown in Lambert Conformal Conic projection. It has been zoomed in to show only a few of the highly Gerrymandered districts.

The drawing of congressional districts was taken from the National Atlas of the United States. The table includes data fields for each district. Some districts consist of more than one area object (typically islands) so there is more than one entry in the drawing's table for each district. If we were creating this drawing for "real life" use instead of as an export example, we probably would have taken a moment to use Dissolve to collapse all of the separate area objects associated with each district into a single area object that is "branched," that is, it looks like different objects but in fact all the islands and other visually separate pieces are part of the same area object. We could use the NAME field to do a dissolve.
We export the drawing to KML by choosing File - Export - Drawing as in the previous examples, choosing a filename ending in .kml to create a KML file. In the Export KML File dialog we choose the NAME field for Name and the URL field for the Description. Press OK.

When we open the KML in Google Earth we see the areas have been rendered with partial transparency. The color of the area border line is black just as was the formatting in the drawing.

**Note:** Manifold can export from maps to KML, in which case the drawing objects sent out to the KML file will be rendered with whatever layer opacity was used for that layer in the map; however, areas are always rendered using partial opacity with that partial opacity being modified by whatever layer opacity setting has been used in a map. That is, areas can be made even more transparent but can not be made fully opaque.
If we zoom into the region in the lower right we can see an example of anti-democratic Gerrymandering, the contorted blue and purple districts. The purple district is the voting area for a single Congressional representative. It is one contiguous area and not several disconnected areas. For example, it is attached by fine corridors passing through enemy territory along the lower left portion of the district. The blue district looks like two or three disconnected regions, but each is indeed connected to the others by thin corridors. This sort of thing passes for democracy in some states, usually accompanied by chants of "let every vote count."

Let's go back to doing GIS. We note that exporting the NAME and URL fields to the Google Name and Description fields added information to the Google Earth Places pane.
We can travel to a desired district by double clicking on the Name next to the pushpin icon.

Google Earth will immediately zoom us to the district for Congresswoman Linda Sanchez in the Los Angeles area.

Google Earth is smart enough to know a URL when it sees one, so it shows the URLs in the Description field as live links. If we double-click on one of them we launch the browser built into Google Earth. We'll click on the Sanchez home page.
...and there we see the Congresswoman herself. Nice pearls!

We can click on the URL for Trent Franks to see his web site as well:

Interesting to see the PhotoShop manipulation of the background scene, which has been extended to the right through mirror repetition. A very nice effect, but like so much in Congress, on both sides of the aisle, not exactly true to reality.

Note: Manifold takes no sides in politics and has no knowledge whatsoever of the politics of the congressional representatives whose districts were picked out by the Gerrymandering rankings. They simply appear because their districts were identified by an automatic formula as especially flagrant examples of Gerrymandering given their perimeter to area ratios. Without any human intervention at all the automatic formula picked out exactly equal numbers of districts from both parties. Gerrymandering is used as an example because GIS is central to Gerrymandering, both as the central tool used to construct biased districts as well as the tool used to reveal them.

Areas Shown as Hatched Lines

Some authors like to show areas using a pattern of hatched lines. This allows a very bright, opaque color to be used that still allows some of the background to be visible. We can achieve this effect by creating a fake area of sorts that is cobbled up from lines arranged into a suitably dense hatch pattern to give the appearance of an area.
This is an advanced method not likely to be used by most people. After all, it’s easiest just to export areas as partially-transparent solid color and be done with it. But Manifold is so good at spatial manipulation and editing that we cannot resist including this method to show how such things are done. What follows is very straightforward and even fun for GIS people but may strike the beginner as annoyingly technical.

This section of the topic assumes the reader is familiar with making selections, using the transform toolbar and understands elementary copy and paste of objects between drawings that are layers in maps.

We've created a map from our drawing of Placer County and have added two new drawing layers, one called Drawing and one called Grid. The Grid layer is the active layer.

We choose View - Grid, check the Show grid box to enable the controls and choose a spacing of 0.01 degrees. We choose a Style of Solid Lines and then press the Create button. After pressing the Create button we uncheck the Show grid box as we don’t want to actually show a grid, we just want to create line objects where the grid would be shown.
A nice dense grid of lines appears in the Grid layer.

Click the Grid layer off so it does not distract. Click the Drawing layer tab so it is active and draw an area in that layer using the Insert Area tool as set forth in the Adding Shapes topic.

Usually, when we want to display an area in Google Earth it is some area that we get by importing a drawing that contains areas. To keep this example visually simple, we'll just create an area freehand using the Insert Area tool. The Placer CA Drawing layer is just being used as a background so we know where we are drawing the area. It doesn't really matter where the area comes from or whether we use a background layer to guide our drawing, as we can still use the techniques in this example regardless of where the areas come from or how many of them there may be.
In the transform toolbar we use the settings shown above with the Clip with (Intersect) operator and press Apply. This uses the area in the Drawing layer to “cookie cut” the grid of lines in the Grid layer. It still works even though the Grid layer has been turned off from view.

We next use the Boundaries operator in the transform toolbar to make a line that follows the edge of the area.

The new boundary line appears in red selection color. We want this line because otherwise the grid of lines in the Grid layer we have just cookie-cut with the previous command will have jagged edges (much as cutting a wire screen leaves pointy wires sticking out along the edge) and we would like to have a nice, neat line around the edges.

Next, with the Drawing layer still the active layer, click onto the area to select it and then press Delete to delete the area object. That will leave just the boundary line. (This step is not illustrated.)
Click on the Grid layer tab to turn on the layer and to make the Grid layer the active layer. We see that the gridlines cut out by the Clip with (Intersect) operator are still selected. Use Edit - Copy to copy these lines.

Click off the Grid layer and then click on the Drawing layer tab to make it the active layer. Choose Edit - Paste to paste the lines copied from the Grid layer into the Drawing layer. Press Edit - Select None to deselect all objects in the layer to get the view above. We've also reduced the opacity of the Placer CA Drawing layer so it does not visually interfere so much.
We change the name of the Drawing layer to Study Area to give it more credibility as a "real world" example. Next, we change the foreground color of lines in the Study Area drawing to a bright color so they will be more visible when viewed in Google Earth.

We open the Study Area drawing in its own window. All the work in the previous steps boils down to this one drawing, a neat section of grid lines cut out to fit an area we wanted to show, surrounded with a neat boundary line that exactly fits the edge of the area. Better still, this is a true GIS drawing that is perfectly georegistered and exists exactly in the region where it was drawn in Placer County.

We cannot create such a drawing in non-GIS tools like Illustrator because such packages do not create georegistered drawings. Without a true, georegistered existence within a real projection the lines could not be displayed in their true location and scale within Google Earth. Without some sort of background map layer to provide a visual context for the above drawing we might not notice it is indeed georegistered unless we pass the mouse cursor over it and see the geographic coordinates reported in the status bar, but it is still nonetheless completely and perfectly georegistered in a way that only a GIS package can do.

Exporting this drawing to KML takes no time at all. As before, with the drawing open we choose File - Export - Drawing, in the Save as type box we choose KML / KMZ Files, provide a suitable file name such as Study Area.kml (don't forget the .kml extension) and press Save.
We now can double-click the Study Area.kml to launch Google Earth to display it.

If Google Earth has been left open with the previous drawing of Placer County lines displayed, double-clicking the Study Area.kml drawing will display both sets of lines at once.

If Google Earth has been closed, then double-clicking the Study Area.kml drawing will launch Google Earth and display just those lines.

Note how precisely the grid lines forming the grid and the boundary line created from the edge of the area fit together. We can zoom however far we like into these lines and they will fit together seamlessly, because they have been built using the extraordinary precision of Manifold.

This example has been written using a single, simple area created freehand but the techniques it shows can be used with "real life" areas that are far more complex.
For example, the area above shows a contour area generated with the Contours command to show regions between 1000 feet and 1200 feet above sea level in the sample Montara Mountain digital elevation data SDTS file. We have been careful to re-project all components into Latitude / Longitude projection to assure a fast export.
Following the procedure given in this topic allows us to show that area in Google Earth by creating a KML file, seen above in Google Earth. That diagonal furrow, by the way, running just to the left of highway 280 is the San Andreas fault.

**Exporting Images to KML**

There's one last tool we have in our arsenal of techniques for exporting data to KML and that is to export an image. Images may be used to show the images themselves for special effects, to represent areas, or even to show surfaces as images for special effects or to display gradient data.

This example shows how to represent areas using an image, continuing with the same map used earlier.
We begin by adding a **Circle** drawing to the map and then we create a circular area using the **Insert Circle** tool. We format the area with yellow background color. Creating the circle in the map allows us to guide the creation of that circle using the background drawing of Placer County as a visual reference.

We open the new drawing in its own window and zoom in to the circle.
We use Tools - Make Image to make an image of suitable resolution.

We open the image in its own window and see it has some white pixels and is larger than necessary.
We crop the image down in size and then select the white pixels.

We delete the white pixels leaving yellow with transparent pixels. We export this image to KML in the usual way. This creates both a KML file and an accompanying .tif file. Both files must be kept together and provided to our friends if they would like to see the image in Google Earth.
We double-click open the KML file and the accompanying .tif is automatically loaded into Google Earth. We've kept a Google Earth session open after previously loading the Study Area.kml file so the contents of the two .kml files can be compared.

**Important:** For this example we've created a KML file so that it is clear how Manifold creates a .kml and an accompanying .tif image file. In "real life," when we export images we use KMZ format, ending the filename in .kmz. Manifold will then create a single “zipped” .kmz file that contains the image within it. This is far preferred to using .kml because then we don’t have to worry about keeping two files (the .tif and the .kml) together for use.

Note that the circle is no longer a circle. That is because drawing a circle in Latitude / Longitude projection results in an ellipse on the actual Earth's surface at the latitude of Placer County, and that is how it is shown using the visual display (a type of projection) employed by Google Earth. Looking carefully at the Study Area shape we can see it too is somewhat different in shape than it appeared in Manifold in Latitude / Longitude projection, as expected.

Note also that just as with drawings, we cannot create such an image in non-GIS tools like PhotoShop because such packages do not create georegistered images. Without a true, georegistered existence within a real projection the image could not be accurately displayed within Google Earth. To prepare images for accurate display within Google Earth we must use a package with real GIS capabilities.

**See Also**

Import a Drawing from the Geocoding Database
A Flashy Demo - Web Queries and KML
Exporting KML to Google Earth
Linked Images from Google Servers
Export Drawing - KML, KMZ
Export Image - KML, KMZ
Import Drawing - KML, KMZ
Display Demographic Data in a Thematic Map

A frequent use of Manifold System is to display data on maps or drawings by coloring areas or points in a range of colors corresponding to the data values. Such a display is said to be thematically formatted and such maps are called choropleth or thematic maps.

**Important:** If you have installed 64-bit Manifold, do this example in 32-bit mode by launching Manifold System using the *Manifold System (32-bit)* shortcut. See the note on 64-bit Manifold at the end of this topic.

There is a huge amount of data available for download by Internet. In the case of the US, Manifold FTP sites and Census Bureau sites provide demographic data for US States, US Metropolitan Statistical Areas, and for the World by country. This data is very useful for the creation of thematic maps and for statistical analysis. In addition, numerous other sources on Internet provide downloads for vast amounts of statistical information.

Demographic data on the Internet is provided mostly in the form of database files, such as Access .mdb or dBase .dbf files. To create maps using this data, we follow a four step process:

- Find and import an appropriate base drawing to display the data (such as the various world, US states or other drawings provided on the CD).
- Import or link the table containing the demographic data of interest.
- Add a relation to the drawing’s table with the demographic table that makes fields of interest from the demographic table appear within the drawing’s table.
- Thematically format the drawing using fields of interest.

For the above process to work, the base drawing and the table containing the demographic data have to share at least one field in common that can be used as a key field. For example, if we want to display demographic data for each state in a drawing of the United States, both the drawing and the data table must each have a field that identifies the state (such as a state name field or two letter postal abbreviation).

In this example, we will use the *States_thematic* map and a table from the *StatesVital.mdb* database. Both of these files are located in the help examples folder on the Manifold CD. The example assumes you are familiar with basic Manifold concepts such as importing drawings and tables, adding relations and applying thematic formatting.

**Step 1:** Import the States_thematic drawing.

We begin by importing the *States_thematic.mfd* drawing into the project using **File - Import - Drawing**. Check all of the boxes to import all of the fields in the **Import MFD File** dialog.

The *States_thematic* drawing shows US States as areas, including a FIPS code for each state.
We can see the fields in this drawing's table by double clicking the table open in a table window.

**Step 2: Import the demographic data table.**

We import the StatesVital.mdb table into the table by choosing File - Import - Table. [If we desire a more dynamic link to the data, we could link the table using File - Link - Table - see the discussion at the end of this topic.]

When we open the StatesVital.mdb file we will see the Import MDB File dialog.

For this example we will import only one table by clicking on Death Rates by Leading Cause and pressing OK.

If we double click open this table after importing it we will see it is a very interesting table providing statistics on death rates from heart disease, cancer, accidents, motor vehicle accidents and many other causes of death for the various states. The rates are per 100,000 resident population in 1994.
So far we have imported a drawing and a table. If we look at the project pane we will see the States_thematic drawing with its associated table as well as the Death Rates by Leading Cause table.

Note that this table includes a FIPS field that contains values that match up to the FIPS field that is in the drawing's table. This is a very important observation, since we will be using the FIPS codes in the thematic states drawing to match the FIPS codes in the demographic data table.

**Step 3: Add a relation to the States_thematic drawing's table.**

Adding a relation to a table is simply adding a column to the table that is taken from a different table. To add a relation to the States_thematic Table, we click it open and then choose Table - Relations.

The Relations dialog for the table opens with no relations, since none have yet been added. We press the New Relation button to add a relation.
The Add Relation dialog asks us which fields will be used to match up records between the States_thematic Table and the new table from which we will bring in a column. In the pane on the left we choose FIPS as the field to use for the States_thematic Table. The combo box over the pane on the right allows us to choose which table will be used to supply fields for this relation. Since we have imported only one other table it is already loaded in the combo box. We choose FIPS in the pane on the right to use as the key field for the Death Rates table. Press OK. Note that it is a happy accident that the key fields on both sides have exactly the same name. The important thing is that the contents of the fields correspond.

Back in the Relations pane we now have a new relation added in the upper pane. This relation uses the FIPS field as a key to bring columns in from the Death Rates table using the FIPS field in that table as a corresponding key. The lower pane shows various cheerful columns available from the Death Rates table that we can bring into our States_thematic Table as relations.

We will check the Accidents and adverse effects rate box as well as the Cancer Rate, Cerebrovascular disease and Homicide (scrolled out of view in the illustration) boxes. Press OK.
Like magic, new columns such as Accidents and adverse effects rate column appear in the States thematic Table. (If you don’t see the columns, use View - Columns to click them on).

We can see the columns better by using View - Columns to hide the other fields in the table and show only the state name field and the new columns. The new columns provide data from the Death Rates demographic table.

**Step 4: Apply thematic formatting to drawing.**

We click onto the drawing to move the focus there.

We will thematically format the drawing by changing the background (“fill”) color of the areas. To do this, click on the area background color well.
In the pull down color menu, choose Theme.

The **Format** dialog launches with default values. We will change the default values by changing the choices in the **Field**, **Method** and **Palette** boxes. We will also check the **Preview** box so we can see a preview of the effects in the drawing window.
We choose the *Accidents* column in the field box and the *Natural Breaks* method. We choose *Reds to Yellows* as our color palette and press *Apply*. The resultant colors are nice, but since we are indicating death rates we would prefer to have redder colors indicate higher values.
Pressing the **Reverse** button reverses the colors so that redder colors are used for higher values. Press **OK** to apply the thematic format.

The resulting drawing shows how various states compare in terms of accidental death rates (excluding motor vehicle accidents). It's no surprise that states representing a frontier, outdoors lifestyle such as Alaska, Idaho, Wyoming and Montana have higher accidental death rates than do states with population concentrated in urban areas. However, why the cluster of high accidental death rates in the deep South? Why does New Mexico have a significantly greater rate of accidental deaths than Arizona? Could it be that a higher population of retirees in Arizona than in New Mexico skews mortality statistics away from accidental deaths?

To see the other demographic data we imported we can open the thematic format dialog and change the field used.
For example, we could choose **Cancer rate** as the controlling field and press **Apply** (and then also press **Reverse** to reverse the present color range) to see the states colored by cancer rates.

This is a very interesting drawing. Note that Utah is exceptionally low in cancer rates while the Rust Belt, Appalachia and the southern regions feature high cancer rates. As expected, Florida has high rates as well, perhaps from the high percentage of older residents. We’d bet that this map would be very similar to a map created of smoking rates per 100,000 citizens. Note that Utah has a high percentage of Mormons who as a statistical cohort tend to show a bias against smoking. Both West Virginia and Pennsylvania have high cancer rates, no doubt due to the higher percentage of persons occupied in coal mining in those states. An anomaly: why the higher cancer rates in Maine?

Finally, we will look at death rates by homicide (murder). To do so we open the **Format** dialog one more time and choose **Homicide rate** as the controlling field, press **Apply** and then press **Reverse**.

The **Natural Breaks** algorithm assigns states to intervals based on clustering of homicide rates. The drawing shows a classic observation, that increased homicide rates are correlated with warmer weather. New York state and the District of Columbia (the seat of Federal government in the US, too small to be seen in this zoomed out view) are clear exceptions.
The States_thematic Drawing

For the convenience of users, we have provided the States_thematic drawing that shows Alaska and Hawaii in the same view as the lower 48 US states. Here is how it was created:

1. Project the States map into Lambert Conformal Conic projection using an origin approximately in the center of the US.
2. Move Alaska and Hawaii next to the lower 48 states.
3. Resize Alaska to reduce the size (using the Scale transform) and rotate it counter clockwise about 28 degrees (using the Rotate transform).
4. Resize Hawaii to enlarge it (using the Scale transform) and rotate it about 30 degrees counter clockwise.

This drawing is useless for serious measurement since Alaska and Hawaii have both been re-scaled and moved, but it does provide a handy drawing for presentations. Just make sure to note somewhere that Alaska and Hawaii are not shown to scale!

Using drawings such as the States_thematic drawing which take geographic entities and apply crafty GIS tricks to move them into different locations and scales within what is purportedly a “real” projection is in some respects an amateur hack, beloved though it may be for expediency even by professionals. Sometimes it is best to simply cringe, clench our teeth and just do what works quickly.

A more elegant and rigorous way to accomplish a presentation like that of the States_thematic drawing is to use a print layout that has different views in play as elements in the layout. Even within an IMS application we can code a web page that uses frames to show multiple IMS views of multiple maps that contain the same drawing, but with different views or projections used so that the continental US as well as Alaska and Hawaii can all be present on the same page.

Discussion

This example explored a few fields from one table out of many thousands of fields available. The same methods may be used to explore demographic data downloaded from the Manifold FTP site or from other web sites, such as the Census Bureau's web site. For example, from the Manifold FTP site we can get and explore the WorldData.mdb database of world demographic data in conjunction with the World_demographic drawing. All of the tables for world data are in the one WorldData.mdb database file.

Note: Manifold FTP sites (drill down through the Manifold web site to find them) require an FTP client that can autoswitch between PASV and non-PASV modes. Ordinary browsers usually won't work, so acquire and learn to use a dedicated FTP transfer program. See the notes on the Manifold web site for more about using FTP.

This example imports an Access .mdb table to keep all parts of the example conceptually simple and as transparent as possible for new users. After all, most Microsoft Office power users know all about how to create a new Access table and save it in an .mdb file. It would be just as easy to import or link a table from some other data provider, such as a table within an SQL, Oracle, IBM DB2, MySQL or other more sophisticated database system.

A more advanced approach than importing the table would be to link a table into the project to be able to exploit the dynamic properties of linked tables within the project. See the Importing and Linking Tables topic for information on linked tables. If the table were linked, then any change in the external database file or system that provided the table would be automatically reflected in the linked table. For example, suppose the table came from a SQL Server table that gave the total number of automobile accidents per state for the previous week. If we linked the table and then formed a relation between our drawing's table and the linked table, any change in the external SQL Server table would automatically be updated in the drawing as well.

We could exploit this by, for example, creating a Manifold IMS web site that showed a thematic map of the United States showing the rates of automobile accidents in the preceding week, where the data to color the map would be automatically taken from the originating SQL Server table. As our other programs or user interfaces updated the SQL Server table the Manifold IMS display would always be automatically up to date as well.

Tech Tips
Be alert to possible problems when using fields as key fields to form relations. Tables may store information differently than expected. For example, the FIPS codes are stored as text strings in both the States_thematic Table and the StatesVital database even though they are numbers. The text string "35" is not the same as the text string "00035" or as "35.0" even though humans would read all three text strings as the same numeric value. Manifold, however, would not be able to match the string "35.0" to the string "35" in the automated key field matching process. If your drawings and databases use inconsistent representations change the types of the fields to match. You may also have to use Transform toolbar operations to alter field contents into compatible form.

The States_thematic Table has FIPS codes in the "concatenated" format favored by the US Government for use when States and Counties might be in the same database. These are five character codes where the first two characters give the state FIPS code and the last three characters are significant for the county within the state. Therefore, "23000" is the FIPS code for Maine and "23003" is the FIPS code for a county within the state of Maine.

One often encounters State FIPS codes in non-concatenated form. In such instances, the FIPS code for Maine is 23, not 23000. A further complication is that Federal databases often express the FIPS code as a text string and not as an integer. Finally, Federal databases often utilize FIPS codes for countries that are multiplied by 100 and also contain extra leading "0" characters. All of these issues may arise when one combines data and maps from various sources. Each may be easily dealt with through the use of the Transform toolbar or the Convert Type command in the table column head context menu.

How to produce leading zeros with the Transform toolbar? Suppose we wish to create text "numbers" like "00035." Start with an integer field. Add a large multiple of ten (say, 1,000,000) to that field. Convert the integer field into a text field. Cut off as many starting characters as desired, leaving as many leading zeros as desired.

Important Note when Using 64-bit Manifold Editions

Due to a lack of required Microsoft facilities in 64-bit Windows systems, Manifold in 64-bit mode cannot export, import, export or link to DB, HTML, MDB, XLS or WKx format files. This includes no access to the MDB parts of Manifold MFD and MapInfo TAB imports. The workaround for importing or exporting such files is to launch Manifold in 32-bit mode by using the Manifold System (32-bit) shortcut, perform the export from or import into a .map project file and then re-launch Manifold in 64-bit mode using the Manifold System (64-bit) shortcut. Linking is more complex: the data must be in some format usable in 64-bit mode within 64-bit Windows systems, such as a SQL Server database. Alternatively, the data can be kept within a Manifold .map project file and linked using the Manifold ODBC driver.

See Also

See the Color topic for fast "four-color" coloring of drawings.

Thematic Formatting Example

Thematic Formatting and Labels
Selection in Images

This example topic shows use of selections in an image, using the globe.jpg sample image from the Manifold Release CD. We will use saved selections and selection commands to select only South America. We then apply visual effects only to the land area of South America.

Step 1: Create a new project

Use File - New to create a new project.

Step 2: Import the globe.jpg image

Use File - Import - Image to import the globe.jpg image into the project.

Step 3: Open the image in an image window

Double-click on the image in project view to open it. The system will display an info bar warning that the projection of the image has not been verified. In this example, we are not concerned with the projection of the image, so click the info bar to launch the Assign Projection dialog and click OK in the dialog to confirm that the projection is OK.

Step 4: Increase contrast

We will increase the contrast in the image to make it easier to "touch select" the pixels representing the ocean. Do this by choosing Image - Adjust - Brightness / Contrast and entering a value of 40 for the Contrast. Enter the "40" value either by moving the slider bar or by clicking into the number box and entering 40 from the keyboard. Click OK.
The result is an image with brighter light tones and darker dark tones.

**Step 5: Open the saved selections pane**

We will be saving some of the selections we make. Open the saved selections pane by choosing **View - Panes - Selections** or by pressing the **SHIFT-ALT-S** keyboard shortcut. Drag the saved selections pane to a convenient place on screen where it does not cover South America.

**Step 6: Select oceanic pixels**

Using Select Touch in Replace Selection mode, **SHIFT** - click anywhere on an ocean to select all pixels in the ocean, as well as all other pixels with this color. The **SHIFT** click is an instruction to select all pixels of the touched color, including those pixels not contiguously connected to the touched pixel. This command is necessary because at the resolution of this image there is not a continuous run of pixels between the Red Sea, Black Sea and other oceanic areas. Therefore, simply clicking on an ocean area will not also select the isolated pixels. Using a **SHIFT** - click will select the isolated pixels as well.

The result is that all pixels in oceanic regions are selected.

Note: Using **SHIFT** - click to select all of the oceanic areas at once depends on the color used for oceanic pixels being unique. If the same shade of blue also appeared within continental areas we could not take this approach. Instead, we would use Add to Selection mode and then with Select Touch we would click the main ocean region, the Black Sea, the Red Sea and so on to build up the selection by clicking each contiguous region of pixels having the color we wish to select.

**Step 7: Save oceanic selection**

In the saved selections pane, click the **New Selection** button to add this selection to the saved selections.

**Step 8: Change the saved selection's name to "Oceans"**
In the saved selections pane, double-click into the “Saved Selection” name and change it to "Oceans" and press Enter.

**Step 9: Select overlapping circles to cover South America**

Clear the selection.

With selection mode set to **Add to Selection**, use **Select Circle on Center** to select a circle that covers much of South America.

Repeatedly use **Select Circle on Center** to cover all of South America with overlapping circles.

We don't have to be very precise, since the next step will trim the selection. We can safely overlap ocean areas.
We only need to be sure that the circles do not overlap land areas (such as various islands) we do not wish to select as part of South America.

**Step 10: Subtract the Oceans saved selection**

In the saved selections pane, click on the **Oceans** saved selection and then press the **Subtract from Selection** command button. The pixels in the Oceans saved selection will be subtracted from the current selection created with overlapping circle selections.

The result is a selection that covers only the land area of South America.

Note: had we wished to see what effect subtracting the Oceans saved selection would have before pressing the **Subtract from Selection** button, we could have checked the **Preview** box in the saved selections pane. This would have previewed the Oceans selection in blue color when the Oceans selection was highlighted. This would let us see how it overlaps the existing selection. For examples of using **Preview**, see the Invisible Pixels and Selection and the Modifying Selections topics.

**Step 11: Save the South America selection**

Just in case we would like to later recall just the South America selection, we can save it in the saved selections pane. In the selections pane, click the new selection button to add this selection to the saved selections.
Double-click into it to rename it to "South America."

**Note:** The selections pane can hold up to six saved selections for each image or surface with a seventh selection reserved for invisible pixels. Drawings and labels components can have seven saved selections.

**Step 12:** Alter only the South America selection

We can pull down the Selection Style menu to choose a **Border** style of selection.

This outlines the selected region with a border line in red selection color so that the red selection pattern does not interfere with our view of the selected region.

We can choose **Image - Adjust - Hue / Saturation** to alter the hue of the pixels in the selected region. Changing the hue to **-90** will shift the colors towards greener hues. Press **OK** to apply the change.
The result is that South America is colored in green hues.

We can use the above techniques to alter the rest of the image as desired. For example, to create the image above we selected land areas in North America using overlapping circles, and then in the saved selections pane we used the Subtract from Selection command button to subtract the Oceans selection and to subtract the South America selection. We then changed the hue of the remaining North America selection. We also selected the Oceans selection again by clicking on the Oceans selection in the saved selections pane and choosing Replace Selection. We then reduced the brightness of the ocean regions to make these pixels black. Finally, we deselected everything and applied Image - Effects - Relief to give the continents and mountain ranges a 3D appearance.
Displaying Data in a Gradient Map

In the Display Demographic Data in a Thematic Map topic we saw how Manifold can display data by coloring the objects in drawings. At times we have data that is point data which we would like to display in the form of a gradient map, like that illustrated below, that shows smooth changes from one color to another that reflect the values of data in the map.

Gradient maps are usually created from data points. The task of the GIS software is to smoothly interpolate data between the points and to then color it as we desire. In Manifold, we use surfaces to easily accomplish these tasks.

This example shows how to take a collection of data points and create a gradient map. There are two main parts to this procedure: The first part is creating a surface from a collection of points and then coloring it. This is very fast and easy. The second part is trimming the surface so that it neatly aligns with other drawings or maps we might create. This takes a few more steps.

We begin with a drawing of data points. Each point has one field, that gives the percentage of students who drop out of school before completing high school.
We created this drawing by opening the `Counties98_demog.map` in the `Data\US` folder of the Manifold CD. We deleted all counties in the US except those in the continental, lower 48 states. We then created centroids for those counties and used those points as a drawing of data points.

The points are positioned at approximately the center of each US county. The drawing showing the points has been projected into Lambert Conformal Conic projection to show a more natural view of the US.

If we show the drawing of points together in a map with a drawing of the United States we can see how the data points are distributed throughout the country. (Some data points in Southern California are offshore, apparently the result of data points collected for offshore islands such as Catalina Island.)

**Step 1: Create a Surface**

Creating a surface is easy. Right click on the data points drawing in the project pane and choose **Copy** from the context menu. Next, right click on an empty part of the project pane and choose **Paste As - Surface**.
The Paste As Surface dialog pops up. By default, it offers a pixel size in meters, which we will adjust to a pixel size of 5 x 5 kilometers, for a total image size of 912 x 580 pixels (shown in the lower left corner) and about 2 megabytes. There is no extremely precise reason why we chose this pixel size and resulting surface size…it just seems like a good number considering the density of the points and the size of the United States. We check the Use radius option with a radius of 100 km to constrain the interpolation to a reasonable distance from the available data points.

The result is a surface. We will rename this surface Data Surface. When we open the surface we see the display above, which is something of a disappointment before it is colored with a palette. Note that some regions in the Southwestern part of the US are not filled in by the interpolation. The default interpolation radius and other parameters we chose do not allow the interpolation to entirely fill in this region because points there are too sparse. That's OK: Sometimes it is more important to leave an interpolation unfilled to warn the viewer that data for that region is lacking than it is to form a display that is beautifully filled in but fundamentally deceptive.

Note also that the surface has been created using the same projection as the drawing of points from which it originated. It is precisely georegistered.

**Step 2: Color the Surface**

The grayscale display is so ugly that before we do anything else we will color it using more appealing and informative colors. To color the surface with a palette we choose the View - Display Options dialog.
The dialog opens with default settings not using any palette to color the surface. For now, we will leave the Shading and Autocontrast options checked. Choose Spectrum for the Palette.

Press the Apply button in the toolbar to apply the palette to the Colors pane.
The result is a series of equal intervals colored by the palette colors. We will adjust these by manually clicking into each interval number and editing it to make it a round number.
Examples

The above intervals seem reasonable at first glance, as a casual effort to round the intervals automatically created for the Spectrum palette. However, note that whereas the automatically created intervals were exactly the same our casual rounding to the nearest five or tens place has created uneven intervals. Some intervals are five units apart (such as 20.00 to 25.00) and some intervals are ten units apart (such as 30.00 to 40.00). This is sloppy.

If we were more rigorous and creating this example for a paper for publication and not just as an example of how to operate dialogs, we would probably have used intervals that were exactly the same size. This would have involved some tinkering to get even intervals that were reasonably well rounded. For example, we might have used intervals of 8, 16, 24, 32, and so on, or we might have used 20, 25, 30, 35, 40, 45 and 50.

Press OK to apply the display options to the surface.

The result is that the surface is neatly colored. Dark blue colors indicate regions where most students finish high school. Yellow, orange and red colors indicate regions where very many students fail to finish high school.

If we show the new surface in a map overlaid by the drawing of data points we can see the relationship between the points and the surface created from them. Note how the surface cannot be completed in the Southwest because the allowed interpolation radius does not extend far enough to fill in the gaps between the sparse points in that region.
If we overlay a drawing of the United States (the US_Main.mfd drawing found on the Manifold CD, with area background set to transparent color) we can see that the interpolation extends beyond the geographic borders of the US.

**Step 3: Trim Surface to US Boundaries**

Although at this point we already have a perfectly functioning gradient map, as a matter of neatness and better presentation we will trim the surface so that it does not extend beyond the borders of the US.

We begin by showing a drawing of the United States (the US_Main.mfd drawing) in a map together with the surface.
We select all of the areas in the US_Main drawing clicking on its layer tab in the map and choosing Edit - Select All (or by pressing CTRL-A).

Next, we right click on the US_Main drawing layer tab and we choose Transfer Selection.

In the Transfer Selection dialog we choose Data Surface as the component to modify, using the US_Main drawing. Since this is the only component in our project that has a selection in it, it is the only component that appears in the Using pane. Press OK.
The selection will be transferred to the surface, as we can see by clicking off the US_Main drawing that overlays it.

To trim the surface we choose **Edit - Select Inverse** (or, press `CTRL-I`). This inverts the selection so that all pixels are selected except those that had been previously selected. Choose **Edit - Delete** (or, press the **Delete** key) to delete these pixels.
The result is a surface that has been neatly trimmed to the borders of the US.

To provide some context to our map we can turn on the US_Main drawing again, set the background color of areas in the drawing to transparent color and then change the foreground color to medium gray. We can then change theOpacity of the US_Main drawing to 50%. This will provide a faint overlay of state and US boundaries to provide context for the gradient map underneath.
To complete the map, we can add a legend, editing the values in the legend so that the numbers are not too long. See the Adding a Legend example topic for a step by step procedure for adding a legend.

**Comments**

This example used a collection of data points to create a gradient map. The data points used were US Census data from 1990 that gave the dropout percentage for each US county similar to what is in the counties maps on the CD such as *Counties98_demog.map*. To create the data points we simply took the centroid of each county within the continental US.

This is something of a contrived example that takes advantage of a handy data set. To display demographic data, such as the dropout percentage, that is collected on a per-county basis it might make more sense to use a thematic map like that shown in the Display Demographic Data in a Thematic Map topic.

If we thematically format a map of US counties with the same data and the same palette we can compare the visual effect of the thematically formatted drawing to that of the similarly colored gradient map.
In one sense, the drawing is more accurate because this particular data set was collected for each county. In another sense the gradient map surface is more accurate because human phenomena like populations and educational achievement do not suddenly change at the sharp boundary of a county. They are more likely to smoothly vary from place to place.

Gradient maps are typically used to show data arising from natural phenomena that are fundamentally continuously variable over a geographic region, so that one would see a smooth variation beyond the boundaries of just one county or one state. For example, temperatures, pollution, the percentage of ozone in the upper atmosphere or the distribution of algae in a region of an ocean are all cases where we would expect there to be some smooth variation between data points recorded. The technique described here works with all these cases.

The process is always similar: acquire the data points, noting the latitude and longitude at which each sample was collected. Collect the data points into a geocoded table. Create a drawing of points from the geocoded table and then create a surface and color the surface as desired. If desired, trim the surface neatly to whatever geographic boundaries are convenient for the area of interest.

Note that we could create the surface directly from a geocoded table by copying the table and pasting it as a surface. However, it is usually wise to create a drawing of points from the table first as a way of checking that the points really are in the region we desire (and no systematic error has occurred) by viewing them overlaid upon a drawing showing the area of interest.
Turning Layers Off/On by Zoom

We would often like to create a map where different layers turn on and off as we zoom in and out. This allows us to hide dense layers when zoomed far out and to make them visible when zooming in. This example creates a simple map of Mexico with different layers.

Important: If you have installed 64-bit Manifold, do this example in 32-bit mode by launching Manifold System using the Manifold System (32-bit) shortcut. See the note on 64-bit Manifold at the end of this topic.

We would like to create a map of Mexico that has boundaries, highways and railroads layers plus a drawing layer showing data points and a labels layer.

Without the use of zoom ranges, the map looks very busy when zoomed out.

If you do not have an analogous map of your own to use and wish to follow along in this example, do the following to create a map like the one illustrated in this example:

Step 1: Import the drawings to be used in the map

Drawings used in this example are in the Examples directory of the Manifold CD. Use File - Import - Drawing with file type set to MFD Files to import Mex boundaries.mfd, Mex highways.mfd and Mex railroads.mdb.

This creates three drawing components in the project pane, as seen above.

Step 2: Create a Labels Component

Create a new labels component by clicking on the Create button in the project pane toolbar and choosing a new Labels component (or, by choosing File - Create - Labels from the main menu).
In the Create Labels dialog, click on the Mex boundaries drawing component. This will make its fields appear in the dialog's Columns pane. [For the sake of brevity in this topic henceforth we will refer to the Mex boundaries drawing as simply the "boundaries drawing" and likewise for the other drawings used.]

![Create Labels dialog]

In the Create Labels dialog, double-click on the Place name field to add it to the labels text and press OK.

![Creating labels for the map]

A labels component appears in the project pane, indented under its parent boundaries drawing.

**Step 3: Create a Map**

To create a map, we use File - Create - Map and in the Create Map dialog check the boxes for the components we wish to appear in the map.

![Create Map dialog]

In the illustration above we named the new map Mexico. We also changed the default order of the layers using the Move Up and Move Down buttons so the map will be created with the order of layers seen above.
The result is a new map component in the project pane. We can double click the map to open it in a map window. Use Zoom commands to zoom as desired into the map.

**Step 4: Apply a projection to the components**

Zoom ranges only work with projected components. They do not work with components in Latitude / Longitude that have not yet been projected. The fastest way to project the components in the map is to first apply a projection to the map using the Edit - Assign Projection dialog. Choose the Orthographic projection centered on Latitude 24 and Longitude -102 (about the center of Mexico, the defaults suggested by the Suggest button). There will be little change in the appearance of the map since the Orthographic projection so close to the Equator is visually very similar to Latitude / Longitude coordinates.

To apply the projection to all of the components in the map, first confirm the original projection of each component by using the Assign Projection dialog, and then project each component to the projection of the map by right clicking its layer tab and choosing Project to Map. This will re-project the component into the same Orthographic projection used by the map. Alternately (and more slowly) in the project pane double click open each of the components in turn, choose Edit - Change Projection and set the projection to Orthographic with a Latitude center of 24 and a Longitude center of -102.

**Step 5: Format the Map Layers**

Seen in default formatting, the map is somewhat confusing.

We will make it more comprehensible in two steps: first, we will apply formatting to distinguish different layers by color and second, we will use zoom ranges to turn layers on and off as needed as we zoom. Apply formatting using the format toolbar using the methods described in the Formatting Drawings topic.

Using the format toolbar change the foreground color for lines of the railroads layer to a dotted style using dark blue foreground and white background colors. Leave the highways layer in the default colors. Change the background color for areas in the boundaries layer to your choice of color (we used a beige color in this example).
The illustration above shows a cluster of green points in the center of Mexico. This example uses a points layer for illustrative purposes. To create such a layer, right-click on the highways layer tab in the map and choose **Add - New Drawing**. This will add a new drawing layer (and create the drawing in the project pane) as a layer just above the highways layer. Use the Insert Point tool to add some points in the center of Mexico and then use the format toolbar to change those points to a bright green background color.

**Note:** The Mexico drawings provided in the **Examples** folder on the Manifold CD don't have any points in the line drawings such as the highways or railroads drawings. This is unusual, since most drawings created for transportation analysis have points at the ends of each line to enable them to be used for network analysis. If you use different drawings in the examples, you may find that line drawings are imported with numerous points. These may be easily deleted. See Point Styles and Sizes for examples.

**Step 6: Open individual components and assign zoom ranges**

In the following section we will open each drawing in its own drawing window. Do this by double clicking on the component in the project pane. It will open in a drawing window (or, a labels window for the labels component). Note that we can have a component window open at the same time we see that component as a layer in the map window.

Let's begin by adjusting the zoom range on the **railroads** drawing layer. To do this, we double click open the **railroads** drawing in its own window and then choose **View - Properties** and click the [...] button to the far right of the **Zooms** caption.

This opens the **Zooms** dialog. By default the zoom range displays the drawing from the **Minimum zoom** (lowest value) to the **Maximum zoom** (highest value). The **Current Zoom** number shows the current zoom in the opened
railroads drawing window. It will vary as we zoom in or out of the window. It has no effect in setting zoom ranges except to provide a guide that may be used to estimate which zoom values to enter in the Minimum zoom or Maximum zoom boxes based on the current appearance of the component window.

In the Maximum zoom list box, we will choose the 1:10000000 preset and then press OK to apply this new zoom range to the drawing. This means that the railroads drawing will be visible in the map from any lowest zoom level up to a highest zoom level of one to ten million. At higher (further out) zoom levels the drawing will not be displayed.

If the map is zoomed out farther than one to ten million the railroads layer will disappear. There will be no change visible in the drawing window for railroads since zoom range only affects the visibility of drawings in maps.

We can repeat this process with the other drawings in the map. For example, we can alter the zoom ranges for the highways drawing.

We double-click open the highways drawing in a window. We can then choose View - Properties and click the […] Zooms button to set zoom range for the highways.

By default the Minimum zoom and Maximum zoom boxes offer preset zoom ranges as factors of ten. We can also enter a zoom range that is not an even factor of ten.

For the Minimum zoom range we can choose 1:1000000. For the Maximum zoom range we can enter 3000000 to specify a zoom range of 1:3000000. These values mean that the highways layer will be displayed in maps at zoom levels between 1:1000000 and 1:3000000. At zoom levels below 1:1000000 or zoom levels above 1:3000000 the highways layer will not be displayed in maps. Applying this zoom range to the drawing instantly
causes the highways to disappear from the map if the map window is zoomed farther out than one to three million.

Next we can click open the points drawing layer and set the zoom range for the points.

We will set it from any lowest zoom up to a highest Maximum zoom value of 1:1000000. In the map window, the data points layer immediately disappears if the map window is zoomed higher than 1:1000000.

Finally, we can click open the labels component and set the zoom range for the labels.

We will leave the zoom range for Minimum zoom blank so there is no minimum zoom value and we will set the Maximum zoom to 1:5000000.

Within the map window we can now zoom in and out and the drawing layers will appear as directed by the zoom ranges we have specified.
Suppose we begin with the zoomed out view seen above. The view is zoomed out farther than 1:10000000 so no layers except the boundaries appear. As we zoom in, the different layers will appear. We will zoom in to the central part of the illustration in each case.

As we zoom in closer than a scale of 1:10000000 the railroads appear in dotted lines.
As we zoom closer than 1:5000000 the labels layer appears. Labels, surfaces and image components can have zoom ranges specified as well as drawings. Assigning different zoom ranges to different labels layers is an essential method to keep maps from becoming cluttered with labels that are too close together.

As we zoom to below 1:3000000 the highways layer appears.
Finally, as we zoom to below 1:1000000 the highways disappear, leaving only the railroads, labels and data points layers.

**Notes**

What does a scale of 1:1000000 (one to one million) really mean? It means that what is shown in one unit of measure on the computer screen is one million of those units on the surface of the Earth. If we measured a feature on the screen that was five centimeters wide it would be five million centimeters (50 kilometers) wide on Earth.

**Important Note when Using 64-bit Manifold Editions**

Due to a lack of required Microsoft facilities in 64-bit Windows systems, Manifold in 64-bit mode cannot export, import, export or link to DB, HTML, MDB, XLS or WKx format files. This includes no access to the MDB parts of Manifold MFD and MapInfo TAB imports. The workaround for importing or exporting such files is to launch Manifold in 32-bit mode by using the Manifold System (32-bit) shortcut, perform the export from or import into a .map project file and then re-launch Manifold in 64-bit mode using the Manifold System (64-bit) shortcut.

Linking is more complex: the data must be in some format usable in 64-bit mode within 64-bit Windows systems, such as a SQL Server database. Alternatively, the data can be kept within a Manifold .map project file and linked using the Manifold ODBC driver.
Create a Table and Add Records

This is a small example showing the mechanics of creating a new table and adding records. Manifold has great table and database capabilities. We will often create small Manifold tables as a means of keeping track of things, almost like a GIS personal information manager. Some users will actually pop open tables in Manifold to keep track of logins for emails, the configuration of their systems and other useful information that has nothing to do with GIS or mapping.

In this example we create a small table and enter a series of records. The records will be later be used as control points to georegister an image.

**Step 1: Create the Table**

To create the table, right click into the project pane and choose **Create - Table** from the context menu that pops up.

In the **Create Table** dialog we press the **Add Column** button to create new fields and then for each new field created we double click into the **Name** and **Type** cells to specify the name desired and the data type of that field. Press **Enter** when done specifying the name or choosing the type to accept the choice into the **Name** or **Type** cell for each field.

We create three new columns, called **Longitude** (of type Longitude), **Latitude** (of type Latitude) and **Name** (of type variable length ANSI text). Pressing **OK** in the dialog creates the table.

**Step 2: Add Records to the Table**

The new table is empty. We double-click into the **Longitude** cell for the first, asterisked record to create a new record.
We can then enter in the value for the longitude for the first record. The default display format for latitudes and longitudes in Manifold is decimal degrees. However, Manifold is smart enough to understand almost any reasonable format in which we enter such values. For example, it knows that if we enter \(-122 20\) where the two numbers are specified by a space we mean the values as 122 degrees, 20 minutes West longitude. An entry like \(-122 20 10\) would be interpreted as 122 degrees, 20 minutes and 10 seconds West longitude.

We can press \(\text{Enter}\) to accept the longitude value. The row handle changes to a pencil icon to indicate we are editing the record. We can double-click into the \(\text{Latitude}\) cell for the row and enter \(37 50\) to mean 37 degrees 50 seconds of latitude. Pressing \(\text{Enter}\) accepts the edit into the cell. To create the record we press the row handle.

This creates a new record using the edit values and adds another row with the new row asterisk record handle. Note that when the new record is created it will be displayed in the format specified for that column. Because the default format for latitude and longitude types is decimal degrees, the values in the cells are now displayed in the decimal degree equivalents to the degrees, minutes and seconds notation we used to enter the values.

To add another record we can double click into the \(\text{Longitude}\) cell for the new record.
We add a longitude value and a latitude value and then click on the row handle.

This creates the new record and moves the new record handle down one position.

At any time we can double click into the Name cells for the records to add values there. At any time we can double click into any of the cells to edit the values if desired.

Notes

It’s easy to create new tables in Manifold. After creating a new table we can add records to it or edit existing values. People will often use Manifold tables as personal database managers. These can be saved in a .map file or exported as desired to database formats such as .mdb or .dbf.

Because Manifold can accept different formats for latitudes and longitudes when entering values, we can use tables as a way of converting from degrees, minutes and seconds format into decimal degrees format or vice versa. At any time we can change the format of the column and see the values in different format.

If we wanted to use the table created above for control points, we could create a drawing from it by copying the table and pasting it as a drawing. See the Create a Map from a Geocoded Table example for the step-by-step procedure for doing so.

Tech Tip

Manifold can read so many different database formats that we can use the system as a converter between different database formats. For example, we can read tables from Access 2000 .mdb format and then export them to Access 97 .mdb format. Or, we could read tables in .dbf format and export them in .mdb format.

See Also
Zoom Ranges
View - Properties - Zooms
**Edit a Table with the Transform Toolbar**

It is often the case that a table contains data in a form that we would like to change so it can be more useful. A frequent task is changing coordinates from various legacy formats to modern digital formats.

In this example, we will use the transform toolbar to change the coordinates in a table from old-fashioned latitude and longitude coordinates as seen in the table above to modern decimal degrees as seen in the table below.

<table>
<thead>
<tr>
<th>FEATURE</th>
<th>TYPE</th>
<th>LATITUDE</th>
<th>LONGITUDE</th>
<th>INDEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abbott Seamount</td>
<td>seamount</td>
<td>31 48N</td>
<td>174 16E</td>
<td>131.75</td>
</tr>
<tr>
<td>Abington Reef</td>
<td>reef</td>
<td>18 00S</td>
<td>149 45E</td>
<td>418.50</td>
</tr>
<tr>
<td>Abre Canyon</td>
<td>canyon</td>
<td>17 31N</td>
<td>120 22E</td>
<td>117.12</td>
</tr>
<tr>
<td>Abraham Canyon</td>
<td>canyon</td>
<td>52 37N</td>
<td>172 20E</td>
<td>152.73</td>
</tr>
<tr>
<td>Abrolhos Bank</td>
<td>bank</td>
<td>18 30S</td>
<td>36 45W</td>
<td>318.03</td>
</tr>
<tr>
<td>Abrolhos Seamounts</td>
<td>seamounts</td>
<td>17 20S</td>
<td>36 30W</td>
<td>317.03</td>
</tr>
<tr>
<td>Accomac Canyon</td>
<td>canyon</td>
<td>37 46N</td>
<td>74 02W</td>
<td>237.04</td>
</tr>
<tr>
<td>Ais Shoals</td>
<td>reefs</td>
<td>3 45N</td>
<td>112 39E</td>
<td>103.13</td>
</tr>
<tr>
<td>Acor Bank</td>
<td>bank</td>
<td>38 10N</td>
<td>29 20W</td>
<td>238.02</td>
</tr>
<tr>
<td>Adak Basin</td>
<td>basin</td>
<td>51 05N</td>
<td>177 00W</td>
<td>251.77</td>
</tr>
<tr>
<td>Adal Canyon</td>
<td>canyon</td>
<td>51 25N</td>
<td>177 05W</td>
<td>251.77</td>
</tr>
<tr>
<td>Adams Bank</td>
<td>bank</td>
<td>41 20S</td>
<td>176 00E</td>
<td>441.76</td>
</tr>
</tbody>
</table>

In this example, we will use the transform toolbar to change the coordinates in a table from old-fashioned latitude and longitude coordinates as seen in the table above to modern decimal degrees as seen in the table below.

<table>
<thead>
<tr>
<th>FEATURE</th>
<th>LATITUDE</th>
<th>LONGITUDE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abbott Seamount</td>
<td>31.800</td>
<td>174.300</td>
</tr>
<tr>
<td>Abington Reef</td>
<td>-18.000</td>
<td>149.750</td>
</tr>
<tr>
<td>Abre Canyon</td>
<td>17.517</td>
<td>120.367</td>
</tr>
<tr>
<td>Abraham Canyon</td>
<td>52.617</td>
<td>172.333</td>
</tr>
<tr>
<td>Abrolhos Bank</td>
<td>-18.500</td>
<td>-36.750</td>
</tr>
<tr>
<td>Abrolhos Seamounts</td>
<td>-17.333</td>
<td>-36.500</td>
</tr>
<tr>
<td>Accomac Canyon</td>
<td>37.767</td>
<td>74.033</td>
</tr>
<tr>
<td>Ais Shoals</td>
<td>3.750</td>
<td>12.450</td>
</tr>
<tr>
<td>Acor Bank</td>
<td>30.167</td>
<td>-25.333</td>
</tr>
<tr>
<td>Adak Basin</td>
<td>51.083</td>
<td>-177.000</td>
</tr>
<tr>
<td>Adal Canyon</td>
<td>51.417</td>
<td>-177.083</td>
</tr>
<tr>
<td>Adams Bank</td>
<td>-41.333</td>
<td>176.000</td>
</tr>
</tbody>
</table>

Tables that use decimal degrees for coordinates are geocoded tables and may be immediately used to create drawings.

This example uses a long sequence of very simple steps. It shows a typical use of the transform toolbar to edit tables using temporary columns as a "scratchpad" to save intermediate results. While it is certainly possible for an expert user to simplify this sequence by using Active Columns or a script, for many one-time tasks it is quicker to simply do conversions manually through a sequence of steps than it is to conceive and debug a programmatic conversion.

The table we will be working with comes from the NOAA's "Ocean Relief" CD, which was published in the late 1990's and provides bathymetric data (underwater terrain elevation) for the entire world. The CD also contains a gazetteer, a listing of underwater features with their types, names and locations contained within a table called **BGNGAZ** (apparently a contraction of "background gazetteer").
The table shows latitude and longitude values using degrees and minutes notation (with a space character between the degrees and minutes) and uses a postfix character instead of modern positive and negative numbers to show latitude and longitude quadrants. This is easy to convert.

**Step 1: Prepend "-" Prefixes to South Latitudes**

The table indicates North or South longitudes and East or West latitudes using "N", "S", "E" and "W" postifixes. Our first task is to prepend "-" characters to South latitudes and West longitudes. We may then eliminate the letter postfixes.

We begin by hiding all columns that are not of interest, such as the **TYPE** and **INDEX** columns.

Next, we create a Unicode text column called **tmp** that will be used to store intermediate results.

Using the query toolbar we select all latitude values ending in **S**.
The illustration above shows the table after unnecessary columns have been hidden, the tmp column has been added and records with latitudes ending in S have been selected.

We use the transform toolbar to prepend a "-" character to all selected values in the LATITUDE column. Note that the transform toolbar is auto-scoped, that is, if there is a selection the action of the transform toolbar is automatically applied only to selected records.

The result is that we have prepended a "-" character to all South latitudes.

**Step 2: Prepend "-" Prefixes to West Longitudes**

We use a similar procedure to prepend a "-" character to West longitudes.

We begin by using the query toolbar to select all longitude values ending in W.

Records with West longitude values will be highlighted.

We can then use the transform toolbar to prepend a "-" character to the LONGITUDE values of all selected records.
Because the transform toolbar is auto-scoped only records that are selected will be altered.

**Step 3: Delete "NSEW" Postfixes**

We no longer need the "NSEW" postfix characters and can delete them. We begin by deselecting all records, since we want the following operations to apply to all records.

No records are selected.

Using the transform toolbar we delete the rightmost single character from all records in the **LATITUDE** column.
This step depends upon our observation that the last character in the LATITUDE column values is the postfix "NSEW" character. In some cases, we may be working with a table with unknown characteristics, where there might be a space character or other "white space" character to the left or right of the value. In that case, it would be wise to use a TRIM transform operator to make sure that the postfix character is indeed the last character (remember, white space characters like a space character are still characters).

Using the transform toolbar we delete the rightmost single character from all records in the LONGITUDE column.

We have now successfully converted all latitude and longitude values from "NSEW" postfix notation to ".-" prefix notation for South latitudes and West longitudes.

**Step 4: Convert Degrees and Minutes to Decimal Degrees**

Our next step is to convert the old-fashioned degrees and minutes values into modern decimal degrees form.

We begin by adding a floating-point numeric column called minutes. This will be used as a temporary workspace column when converting text values into numeric values.
The column appears to the right of the previously created `tmp` column.

```
<table>
<thead>
<tr>
<th>trp</th>
<th>Copy Last Token from</th>
<th>LONGITUDE</th>
</tr>
</thead>
</table>
```

Next, we copy the last token from the `LONGITUDE` column to the `tmp` column. If default settings are in play for token delimiters, the last token is simply the two characters after the space in the longitude values.

<table>
<thead>
<tr>
<th>FEATURE</th>
<th>LATITUDE</th>
<th>LONGITUDE</th>
<th>trp</th>
<th>minutes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abbott Seamount</td>
<td>31 18</td>
<td>174 18</td>
<td>18</td>
<td>0</td>
</tr>
<tr>
<td>Abington Reef</td>
<td>-18 00</td>
<td>139 45</td>
<td>45</td>
<td>0</td>
</tr>
<tr>
<td>Abra Canyon</td>
<td>17 31</td>
<td>120 22</td>
<td>22</td>
<td>0</td>
</tr>
<tr>
<td>Abrahm Canyon</td>
<td>52 37</td>
<td>172 20</td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td>Abrolhos Bank</td>
<td>-18 30</td>
<td>-38 45</td>
<td>45</td>
<td>0</td>
</tr>
<tr>
<td>Abrolhos Seamount</td>
<td>-17 20</td>
<td>-36 30</td>
<td>30</td>
<td>0</td>
</tr>
<tr>
<td>Accomac Canyon</td>
<td>37 46</td>
<td>-74 02</td>
<td>02</td>
<td>0</td>
</tr>
<tr>
<td>Acx Shoals</td>
<td>3 45</td>
<td>112 39</td>
<td>39</td>
<td>0</td>
</tr>
<tr>
<td>Acxu Bank</td>
<td>30 10</td>
<td>-29 20</td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td>Adal Basin</td>
<td>51 05</td>
<td>-177 00</td>
<td>00</td>
<td>0</td>
</tr>
<tr>
<td>Adal Canyon</td>
<td>51 25</td>
<td>-177 05</td>
<td>05</td>
<td>0</td>
</tr>
<tr>
<td>Adams Bank</td>
<td>-41 20</td>
<td>176 00</td>
<td>00</td>
<td>0</td>
</tr>
</tbody>
</table>

Note how this operation copies the two-character token representing the minutes values from the longitudes into the `tmp` column.

```
<table>
<thead>
<tr>
<th>minutes</th>
<th>Copy from</th>
<th>tmp</th>
</tr>
</thead>
</table>
```

The `Copy Last Token from` operator is only available when copying text information between text columns. We used it to get the minutes portion of the longitude text values isolated in the `tmp` text column. We can now copy the values from the `tmp` text column into the `minutes` numeric column and they will be automatically converted into numbers.

```
<table>
<thead>
<tr>
<th>FEATURE</th>
<th>LATITUDE</th>
<th>LONGITUDE</th>
<th>trp</th>
<th>minutes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abbott Seamount</td>
<td>31 18</td>
<td>174 18</td>
<td>18</td>
<td>18</td>
</tr>
<tr>
<td>Abington Reef</td>
<td>-18 00</td>
<td>139 45</td>
<td>45</td>
<td>45</td>
</tr>
<tr>
<td>Abra Canyon</td>
<td>17 31</td>
<td>120 22</td>
<td>22</td>
<td>22</td>
</tr>
<tr>
<td>Abrahm Canyon</td>
<td>52 37</td>
<td>172 20</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>Abrolhos Bank</td>
<td>-18 30</td>
<td>-38 45</td>
<td>45</td>
<td>45</td>
</tr>
<tr>
<td>Abrolhos Seamount</td>
<td>-17 20</td>
<td>-36 30</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>Accomac Canyon</td>
<td>37 46</td>
<td>-74 02</td>
<td>02</td>
<td>2</td>
</tr>
<tr>
<td>Acx Shoals</td>
<td>3 45</td>
<td>112 39</td>
<td>39</td>
<td>39</td>
</tr>
<tr>
<td>Acxu Bank</td>
<td>30 10</td>
<td>-29 20</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>Adal Basin</td>
<td>51 05</td>
<td>-177 00</td>
<td>00</td>
<td>0</td>
</tr>
<tr>
<td>Adal Canyon</td>
<td>51 25</td>
<td>-177 05</td>
<td>05</td>
<td>5</td>
</tr>
<tr>
<td>Adams Bank</td>
<td>-41 20</td>
<td>176 00</td>
<td>00</td>
<td>0</td>
</tr>
</tbody>
</table>
```

Although the values look almost the same, the values in the `tmp` column are text and those in the `minutes` column are numbers. Note that leading zeros are lost in the conversion from text to numbers. Numbers may be divided, multiplied and otherwise processed in a way that text values cannot.

```
<table>
<thead>
<tr>
<th>minutes</th>
<th>Divide by</th>
<th>60</th>
</tr>
</thead>
</table>
```

We can use the transform toolbar to divide the minutes values by 60.
This converts the whole minutes values into decimal fractions of degrees. However, we don't need so many digits after the decimal point. We can right click on the column head of the minutes column, choose Format and then change the format so that only three digits after the decimal point are displayed.

This provides a cleaner display with a significant number of digits consistent with the two-digit minutes values originally used.

We now copy the new decimal degrees values back to the tmp column to automatically convert them back into text form.

The result is a table that has text values for decimal degree fractions in the tmp column. We will "glue" these onto the LONGITUDE whole degree values with a few more operations.
We delete the leading zero in the tmp text values using a Delete Left operator.

<table>
<thead>
<tr>
<th>FEATURE</th>
<th>LATITUDE</th>
<th>LONGITUDE</th>
<th>tmp</th>
<th>minutes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abbott Seamount</td>
<td>31.48</td>
<td>174.18</td>
<td>.300</td>
<td>0.300</td>
</tr>
<tr>
<td>Abington Reef</td>
<td>-18.00</td>
<td>-149.45</td>
<td>.750</td>
<td>0.750</td>
</tr>
<tr>
<td>Abrac Canyon</td>
<td>17.31</td>
<td>120.22</td>
<td>.367</td>
<td>0.367</td>
</tr>
<tr>
<td>Abraharn Canyon</td>
<td>52.37</td>
<td>172.20</td>
<td>.333</td>
<td>0.333</td>
</tr>
<tr>
<td>Abrolhos Bank</td>
<td>-18.30</td>
<td>-36.45</td>
<td>.750</td>
<td>0.750</td>
</tr>
<tr>
<td>Abrolhos Seasounts</td>
<td>-17.20</td>
<td>-36.30</td>
<td>.500</td>
<td>0.500</td>
</tr>
<tr>
<td>Acconac Canyon</td>
<td>37.46</td>
<td>-74.02</td>
<td>.033</td>
<td>0.033</td>
</tr>
<tr>
<td>Ave Shoals</td>
<td>3.45</td>
<td>112.39</td>
<td>.650</td>
<td>0.650</td>
</tr>
<tr>
<td>Acor Bank</td>
<td>38.10</td>
<td>-25.20</td>
<td>.333</td>
<td>0.333</td>
</tr>
<tr>
<td>Adal Basn</td>
<td>51.05</td>
<td>-177.00</td>
<td>.080</td>
<td>0.000</td>
</tr>
<tr>
<td>Adal Canyon</td>
<td>51.25</td>
<td>-177.05</td>
<td>.083</td>
<td>0.083</td>
</tr>
<tr>
<td>Adams Bank</td>
<td>-41.20</td>
<td>176.00</td>
<td>.000</td>
<td>0.000</td>
</tr>
</tbody>
</table>

This makes ready the tmp text values for appending to the longitude data.

We use the Delete Last Token operator to eliminate the minutes values in the longitudes that we no longer need.

Although the last two digits are gone, there is still an invisible space character at the end of each of the text values in the LONGITUDE column.

We can eliminate the unwanted space with a Trim operation.
The table is now ready to append the values in `tmp` to the values in `Longitude`.

This is easily accomplished with an Append operation in the transform toolbar.

The result is a table that has had longitude values converted from an old-fashioned form into modern decimal degrees notation.

The procedure to convert the values in the `LATITUDE` column is exactly the same. Let's take a look at the first few steps:

We copy the last token from `LATITUDE` to `tmp`.

```plaintext
<table>
<thead>
<tr>
<th>FEATURE</th>
<th>LATITUDE</th>
<th>LONGITUDE</th>
<th>tmp</th>
<th>minutes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abbott Seamount</td>
<td>31.48</td>
<td>174</td>
<td>.300</td>
<td>0.300</td>
</tr>
<tr>
<td>Abington Reef</td>
<td>-18.00</td>
<td>149</td>
<td>.750</td>
<td>0.750</td>
</tr>
<tr>
<td>Abra Canyon</td>
<td>17.31</td>
<td>120</td>
<td>.367</td>
<td>0.367</td>
</tr>
<tr>
<td>Abrahm Canyon</td>
<td>52.37</td>
<td>172</td>
<td>.333</td>
<td>0.333</td>
</tr>
<tr>
<td>Abrolhos Bank</td>
<td>-18.30</td>
<td>-36</td>
<td>.750</td>
<td>0.750</td>
</tr>
<tr>
<td>Abrolhos Seamounts</td>
<td>-17.20</td>
<td>-36</td>
<td>.500</td>
<td>0.500</td>
</tr>
<tr>
<td>Accomac Canyon</td>
<td>37.46</td>
<td>-74</td>
<td>.033</td>
<td>0.033</td>
</tr>
<tr>
<td>Alice Shoals</td>
<td>3.45</td>
<td>112</td>
<td>.650</td>
<td>0.650</td>
</tr>
<tr>
<td>Acer Bank</td>
<td>38.10</td>
<td>-29</td>
<td>.333</td>
<td>0.333</td>
</tr>
<tr>
<td>Adal. Basin</td>
<td>51.05</td>
<td>-177</td>
<td>.083</td>
<td>0.083</td>
</tr>
<tr>
<td>Adal. Canyon</td>
<td>51.25</td>
<td>-177</td>
<td>.083</td>
<td>0.083</td>
</tr>
<tr>
<td>Adams Bank</td>
<td>-41.20</td>
<td>176</td>
<td>.000</td>
<td>0.000</td>
</tr>
</tbody>
</table>
```
This isolates the text in the token so that it may be copied to a numeric column.

We copy the *tmp* text values into the *minutes* numeric column.

This converts the values into numbers, which can then be divided by 60, copied back to the *tmp* column to be converted back into text, and so forth. The rest of this procedure will not be shown since it is identical to that used for the *LONGITUDE* column.

After completing the procedure for the *LATITUDE* column we can delete the *tmp* and *minutes* temporary columns. The result is a table that lists features using modern decimal degrees. It is a geocoded table that can be used to create drawings.
Here is an example of a drawing where the geocoded table points are labeled using the values in the **FEATURES** column.

**Comments**

This is obviously a long procedure, but it consists of very simple steps that quickly become second nature to users who need to manipulate tables. Note how the different operations that are possible for text types and numeric types were exploited to get the desired result. Note also how selection was used in the beginning of the procedure to operate on only some of the records.

It’s true that a sophisticated user could write a script, or even an Active Column that would programmatically make the desired conversion. However, it is often quicker to simply use a series of transform toolbar operations, where all results are visually obvious at each step, to accomplish the desired transformation than it is to write and debug a script that accomplishes the same result.

When undertaking lengthy procedures, don’t forget to take a moment to save your work from time to time, just in case a mistake is made that cannot be conveniently undone!
Create a Map from a Geocoded Table

In this example we create a map from a geocoded table created in Microsoft Access. Before beginning this topic, please read the Creating Drawings from Geocoded Tables topic.

Important: If you have installed 64-bit Manifold, do this example in 32-bit mode by launching Manifold System using the Manifold System (32-bit) shortcut. See the note on 64-bit Manifold at the end of this topic.

There are two ways of creating drawings from geocoded tables. A simple, straightforward way if the geocoded table has already been imported into the project is to Copy the table and to Paste it as a drawing. When the drawing is created as a result of the Paste process the coordinates for points in the drawing and their data attributes (if specified for transfer) will be copied from the table. This example shows that process.

Another way of creating a drawing from a geocoded table is to link the drawing to a geocoded table, which might be in the project or might be an external, geocoded table. In this case every time the linked drawing is opened or refreshed it will be dynamically recreated on the fly based on the data in the table from which it is linked. The position of points and their data attributes will be dynamically taken from the table. If the table is altered to add, delete or change the coordinates of records then the points in the drawing will appear, disappear or move as the table records are updated. For an example of this process see the Create a Linked Drawing from a Geocoded Table topic.

The distinction between the above two scenarios is subtle, but critically important. In the first case the spatial data from a table are copied and used to create a new, independent drawing is created. In the second case, a "virtual" drawing is created that remains linked to the table and is controlled by it.

Let's begin this example by considering a geocoded table in an Access .mdb file.

The illustration above shows our table in Microsoft Access. The table is called Waypoints and is saved in an Access .mdb file called waypoints.mdb. There are three fields in this table, called Waypoint, Latitude and Longitude. The table represents a list of GPS waypoints that were measured with a Garmin GPS receiver.

The three fields in this Access table are all text fields. Each waypoint has a name given in the waypoint field. In this example each waypoint has a number for a name, but there is no reason that the text values in the waypoint field could not have been something like "Position A", "Position B", "Bridge", "Buoy 13", "North East Oil Well" or any other string value. In this example they are simply "1", "2" and so on.

For the sake of this example, it doesn't matter how we got the GPS waypoints into a table in Access. Perhaps we entered them into Access by hand while looking at the GPS receiver's readout, or perhaps we downloaded them using software, such as some freeware GPS waypoint download package. The point is that now we have a geocoded Access table that we would like to use in a Manifold map.

Step 1: Launch Manifold and create a new project

Use File - New to create a new project.
**Step 2: Import the waypoints.mdb table**

Use **File - Import - Table** to import the table into the project.

In the **Import** dialog we choose .mdb files in the **Files of type:** box and choose the **waypoints.mdb** database to open. Press **Open**.
The secondary import dialog will show a list of tables that are available in the database. There is only one table in this database file, so we choose the **Waypoints** table and press **OK**.

A new table, called **Waypoints**, will appear in our project pane as a result of the import process. Moving the mouse cursor over the table will show a Windows tooltip that indicates what type of table it is. In this case, we see that it is a **Local** table, that is, it is a table that is stored within the project file and is not linked from some external file or data source.

If we are curious as to what this table contains, we can double-click on this table to open it in a table window.

If we double-click open the table we can see that it contains the fields and records imported from the Access table.

**Note:** The above procedure imports the table into the project. If desired, we could have linked the table into the project by using **File - Link - Table** to link an external table into the project.

**Step 3: Copy the table and Paste As a drawing**

Click on the project pane to move the focus there, and click on the **Waypoints** table to highlight it if it is not already highlighted.

Choose **Edit - Copy** to copy the table to the Windows clipboard, or (as illustrated above) **Right Click** onto the table and choose **Copy** in the context menu. Experienced Windows users will know they can click on the table to highlight it and then choose **CTRL-C** as the Windows keyboard shortcut for **Copy**.
Click on any empty location in the project pane and...

... choose Edit - Paste As - Drawing to paste the table as a drawing.

The Paste As Drawing dialog will appear to ask us which fields we would like to paste from the table, and which fields should be used for the latitude and longitude coordinates. If we have fields in our table that are called "Latitude" and "Longitude," Manifold will make the reasonable guess that these are the fields we wish to use for coordinates and will, for our convenience, pre-load the combo boxes with those choices. If we would like to use other fields, all fields in the table will be available in the choice boxes as well.

A note for experts: unchecking the Latitude / longitude coordinates box will set the coordinate system to the default Orthographic projection. This is the starting point when the X and Y coordinates represent something other than ordinary latitude and longitude coordinates: the drawing is created in Orthographic and then we can assign whatever coordinate system (such as some State Plane projection...) we like.

The Skip zero latitude / longitude records option tells Manifold to skip any records that have empty values for latitude and longitude. Sometimes we will use tables that include erroneous or incomplete records that don't have latitude and longitude values. This might arise, for example, if we have a table of geocoded street addresses where some addresses were not found by the geocoder and thus do not have latitude and longitude values.

In this example, we have checked only the Waypoint field because we don't wish to also import the latitudes and longitudes for each waypoint as data attributes. Manifold will automatically use the latitude and longitude fields to
place each resulting point: we only need check these fields if we also want to have a written copy of them in the drawing’s table’s attributes.

The result of the *Edit - Paste As - Drawing* operation is a new drawing in the project together with its associated table. By default, since there is already a component in the project called *Waypoints* Manifold takes the name of the table that was pasted and appends “2” to create names like *Waypoints 2*. These names can be changed, of course, to something else if desired by right clicking on the drawing and choosing *Rename*.

We can double-click open *Waypoints 2* to see how our new drawing looks.

The new drawing shows our waypoints plotted as points in a drawing. The position of each point is given by the latitude and longitude fields for the corresponding record in the table that was pasted.

If we like, we can also double-click open the *Waypoints 2 Table* associated with the drawing to see what data fields are in the table.

The table has an **ID** as well as the **Waypoint** field we checked in the *Paste As* dialog. Every object in a Manifold drawing has a unique object **ID** field that is the key field used to link rows in the table to objects in the drawing. If we do not wish to see this field we can always *Hide* it in the table.

**Step 4: Create a Map**
To create a map using the points drawing we would most likely import another drawing or two to provide a "base map" drawing that shows the geographic context of the points. Otherwise, if we just look at the drawing of points we don't know where it is located.

In this example, we have imported a drawing, called `bay_land_eg`, which shows the San Francisco Bay region as areas. This drawing is on the Manifold CD.

To create a map that contains `bay_land_eg` and the `Waypoints 2` drawings as layers we right click anywhere in an empty location in the project pane and choose Create - Map. In the Create Map dialog we see all of the components in the project from which we can create a map, only two such components in this case. We check both of them to be used and move the `Waypoints 2` drawing to the top position. For lack of any deeper inspiration we will call this new map component `Map`. Press OK.

The result is a new map component in the project. We can double click it open to see how it looks.
If we click open Map we can see that the GPS waypoints in our geocoded table are various locations in the Southern part of San Francisco Bay. We have taken a moment to change the background color of the points in the Waypoints 2 drawing from gray to yellow.

However, for a more visually appealing presentation we can click on the Bay_land_eg Drawing layer and change the color of the land areas to green. We can then choose View - Properties for the map and change the background color of the map to a custom background color, a shade of blue. That's better!

**Comments**

This same process as used above can be used to display any geocoded table as a drawing of points. The example used a table of GPS waypoints, but it could have just as easily been any other geocoded table. For example, we could copy a table that listed radio stations by call sign and the locations of their antennas in latitude and longitude and paste it into a drawing with points at the location of each antenna.

This very simple example had no fields except for the name of the waypoint and the latitude and longitude. We could have pasted many more fields if they were present in the table. For example, if we had a geocoded table of cities where each city had a name, population, country name or other fields in addition to the latitude and longitude fields we could have pasted such a table as a drawing. The table created for that drawing would include whatever fields we checked in the Paste As dialog during the Paste As operation.

If the geocoded table does not have latitudes and longitudes in the form shown above it is not yet geocoded. Many different formats are used to write down latitudes and longitudes. For example, on occasion one might encounter the coordinates of waypoint 1 above in the form N 37 41.750 W 122 20.380, which is intended to mean North 37 degrees and 41.750 minutes latitude and West 122 degrees 20.380 longitude. If such coordinates
appear for each record it is not yet geocoded. These coordinates must be converted into standard decimal degrees before the table can be used as a geocoded table.

**Tech tip:** Although it is true that Manifold will attempt to understand coordinate values (that is, latitude and longitude values) written in degrees, minutes and seconds notation when pasting a geocoded table as a drawing, as a practical matter due to the wide variation of styles for writing coordinates in such notation (some such styles being utterly unfathomable to humans, let alone to computers) it is wise to first convert latitudes and longitudes into decimal degrees notation, to be sure Manifold can understand. This documentation is therefore written to require decimal degrees notation. If someone disregards that advice and Manifold manages to make sense of the degrees, minutes and seconds notation they are using, that should be regarded as unexpected good luck.

**Changing Ordinary Fields in a Drawing's Table does not Move Objects**

Recall that there are two ways of creating drawings from geocoded tables: one method is shown in this example, to copy the table and paste as a drawing. This creates an independent drawing that has no connection with the table. The second way of creating a drawing is to link the drawing to an external, geocoded table.

In either case, the drawing is created with a table of its own that hosts whatever data attributes are associated with the drawing. When pasting a drawing from a table, we may choose to include latitude and longitude fields from the table. It is important to understand that such latitude/longitude values that occur as data attributes in the drawing’s table **cannot** be used to "move" points. They are simply additional copies in written form of the data that was used to create the drawing and are not the controls that define the drawing.

The positions of objects in drawings are embedded within the geometry of the drawing. They are an intrinsic, built-in part of the drawing that are not listed in the drawing's table except in the form of normally-hidden intrinsic fields in the table. If the drawing has been pasted from a table, the moment the drawing was created the coordinates for each point in the drawing were taken from the table and embedded into the geometry of the drawing and stored within the drawing component. Even if the drawing's table were empty, that geometry would still be inside the drawing. If the drawing is a linked drawing, the geometry is dynamically updated on the fly deep inside Manifold from the table to which the drawing is linked. In either case, the drawing "knows" where the points are located even if there are no coordinate fields (like latitude and longitude) in the drawing's table.

Sometimes we create drawings from geocoded tables that have a latitude and longitude for each record. We might wish to have the latitudes and longitudes that were used to position the points immediately at hand in the drawing's table, perhaps as a handy, human-readable reference of where each point is located. To do so, when pasting the geocoded table as a drawing we may tell Manifold to bring along the latitude and longitude fields along with the other fields, making a copy of the latitude and longitude values to put into the drawing's table.

Whatever the motive or process that was used to create a drawing, it is often the case that "latitude" and "longitude" numbers will be copied to fields in a table as well as being embedded within the geometry of a drawing. GIS beginners often mistakenly think that the positions of items in the drawing are determined from the listings of latitude or longitude fields in the table, so that editing these fields will move the objects. That's not the case.

The position of points in a drawing are determined entirely and exclusively by the geometry information that is embedded within the drawing. Whether the drawing's table has any latitude or longitude fields at all, or whether such values are changed will have no effect on the position of the objects in the drawing.

It is possible to use normally hidden fields inside the table to see the geometry of objects in human-readable form. Manifold has many cool capabilities and one of them is the ability to reach inside a drawing via that drawing's table and to show in human-readable form the geometry of objects. The intrinsic fields in a table display the actual latitudes and longitudes of objects associated with records. However, intrinsic fields are special, system-generated fields and are different than "latitude" and "longitude" fields we might bring into a table that are ordinary text or numeric fields. The rest of this topic explains why this is so.

Drawings are often created in Manifold from geocoded tables. A geocoded table is a table where each record has latitude and longitude fields that specify the location of that record.

Suppose we have an Access database that contains a list of towns together with their latitude and longitude coordinates:
We can import this Access table into a table in Manifold [either by importing the data or by linking to the external table]:

To create a drawing from this table we use Edit - Copy to copy the table and then we use Edit - Paste As - Drawing in the project pane to create a new drawing based on the data in the table.

If our table was called Towns, by default the new drawing will be called Towns 2 and the drawing's table will be called Towns 2 Table. Each geocoded record will appear as a point in the drawing.
We can combine this new drawing in a map together with a base map of the United States to see at a glance where the various points are located.

When we use the Edit - Paste As - Drawing command to create a new drawing, Manifold will automatically create the associated table for that drawing. By default, that new table will have at least one field, the object ID field. During the Paste As operation Manifold will give us the option of copying additional fields from the original table into the new data attribute table for the drawing. The dialog presents a list of check boxes for each field in the original table. We can copy any of the fields we desire.

For example, if our geocoded table is a list of customer records containing fields such as name, address, city, state, and so forth together with a latitude and longitude for each record, we might wish to copy the name, address and other informational fields. There is no particular technical reason to copy the latitude and longitude fields (these fields in the original table will be used automatically to position the points in the drawing for each record) but we can do so if for some reason we want to have this information in the table.

If we do copy the latitude and longitude fields into a table associated with the drawing, we will see them in the table window when the table is opened. It is very important to realize that these are just table fields that were copied from the original table. They are not a “handle” into the geometric coordinates that define the position of the points within the drawing’s internal structure. Altering the values will not move the points, nor are the fields even necessary to keep the points “positioned” at the right spots.

<table>
<thead>
<tr>
<th>ID</th>
<th>Longitude</th>
<th>Latitude</th>
<th>Name</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>-119.47</td>
<td>39.02</td>
<td>Incine Village</td>
<td>NV</td>
</tr>
<tr>
<td>51</td>
<td>-82.66</td>
<td>27.56</td>
<td>St. Petersburg</td>
<td>FL</td>
</tr>
<tr>
<td>52</td>
<td>-123.42</td>
<td>46.2</td>
<td>Lake Oswego</td>
<td>OR</td>
</tr>
<tr>
<td>53</td>
<td>-76.25</td>
<td>40.17</td>
<td>Lancaster</td>
<td>PA</td>
</tr>
<tr>
<td>54</td>
<td>-77.06</td>
<td>38.91</td>
<td>Alexandria</td>
<td>VA</td>
</tr>
<tr>
<td>55</td>
<td>-74.34</td>
<td>39.19</td>
<td>Morgantown</td>
<td>NJ</td>
</tr>
<tr>
<td>56</td>
<td>-70.84</td>
<td>42.5</td>
<td>Marblehead</td>
<td>MA</td>
</tr>
<tr>
<td>57</td>
<td>-88.56</td>
<td>44.02</td>
<td>Oshkosh</td>
<td>WI</td>
</tr>
<tr>
<td>58</td>
<td>-104.79</td>
<td>41.15</td>
<td>Cheyenne</td>
<td>WY</td>
</tr>
<tr>
<td>59</td>
<td>-97.67</td>
<td>30.52</td>
<td>Round Rock</td>
<td>TX</td>
</tr>
<tr>
<td>60</td>
<td>-122.11</td>
<td>47.67</td>
<td>Redmond</td>
<td>WA</td>
</tr>
<tr>
<td>61</td>
<td>-97.34</td>
<td>37.59</td>
<td>Wichita</td>
<td>KS</td>
</tr>
<tr>
<td>62</td>
<td>-122.15</td>
<td>37.45</td>
<td>Palo Alto</td>
<td>CA</td>
</tr>
</tbody>
</table>

If we checked the latitude and longitude fields to be included in the drawing’s table when we pasted the drawing, they will appear in the drawing’s table as seen above. But these latitudes and longitudes are just copies of the data that was used to create the drawing. They do not define the drawing.
For example, in the Table window we could delete the entire "Longitude" column, and the points in the drawing will still be there.

We could delete the "Latitude" field as well. If we do so the points do not magically disappear from the drawing. They will still be there. All that we have done is delete a column from the table associated with the drawing. The points in the drawing have already been created when Manifold read the original database. Changing the text data in the table (either the original one or the new table created with the drawing) changes nothing about those points and does not alter the coordinate numbers inside the drawing.

The above discussion may seem obvious to experienced GIS users; however, for new users it is easy to look at a table full of records with latitude and longitudes and to forget that changing the values in the table does not actually move the points about.

Changing the Location of Points

How then can we change the actual location of points? There are three main ways:

- We can show the (normally hidden) Latitude (I) and Longitude (I) intrinsic fields and then change their values. See Editing Intrinsic Fields in Tables. This is the best method when points must be moved to given coordinates.
- We can right click on individual points in the drawing and use the Object - Coordinates dialog to change the location of the geometric coordinates that define the points. This is a seriously tedious approach.
- We can use any one of a wide variety of interactive editing tools in a map or drawing window to move the points by selecting them and dragging them to a new location.

Note that none of the above options will change the contents of any "latitude" or "longitude" text fields we might
see in a table window. For example if we click and drag a point to a new position using an editing capability that will not change the values of any fields other than the intrinsic fields.

Notes

In this example we created a table using Access. We did this deliberately to show how a geocoded table might look in an application outside of Manifold and then how it is imported into Manifold and used within Manifold.

Some times users who are new to Manifold may think from the use of Access in various examples that Manifold somehow requires the use of Access or is somehow tied to the use of Access .mdb files. That is not the case. Manifold does not need any other DBMS software to function, nor does Manifold require the use of Access or Access .mdb files.

For that matter, we don’t need Access or any other DBMS to create tables in Manifold. We can create tables entirely within Manifold as is set forth in the Create a Table and Add Records example. Manifold includes a built-in database engine of tremendous power and sophistication to facilitate the creation, editing and use of tables, either as standalone tables or as tables that belong to drawings. In addition to that built-in capability, Manifold can also work with virtually any DBMS system or database file format.

This example began with an Access .mdb file because quite often we might encounter geocoded tables in that format. We could have just as easily begun with an Excel .xls file or have given an example using a table from SQL Server or from an Oracle database.

Important Note when Using 64-bit Manifold Editions

Due to a lack of required Microsoft facilities in 64-bit Windows systems, Manifold in 64-bit mode cannot export, import, export or link to DB, HTML, MDB, XLS or WKx format files. This includes no access to the MDB parts of Manifold MFD and MapInfo TAB imports. The workaround for importing or exporting such files is to launch Manifold in 32-bit mode by using the Manifold System (32-bit) shortcut, perform the export from or import into a .map project file and then re-launch Manifold in 64-bit mode using the Manifold System (64-bit) shortcut. Linking is more complex: the data must be in some format usable in 64-bit mode within 64-bit Windows systems, such as a SQL Server database. Alternatively, the data can be kept within a Manifold .map project file and linked using the Manifold ODBC driver.
Create a Geocoded Table from a Map

The previous example showed how to create a drawing from a geocoded table. Sometimes we have a map of points and would like to create a geocoded table from that map.

Important: If you have installed 64-bit Manifold, do this example in 32-bit mode by launching Manifold System using the Manifold System (32-bit) shortcut. See the note on 64-bit Manifold at the end of this topic.

For example, one of the most frequent requests in GIS newsgroups is to get a list of cities with the latitude and longitude of each city together with some data for each city, such as the name, population or other fields. It is often the case that we can acquire a digital map that shows the locations of points such as cities but which does not have the latitude and longitude of each point explicitly set forth in the data attributes. Manifold allows us to use that map to create a geocoded table.

In Manifold, we can create tables that include the latitude and longitude of each point in a drawing by simply copying the drawing and pasting it as a table. This example uses a drawing that shows cities in Europe; however, the same method can be used to create a table with latitude and longitude coordinates (a geocoded table) for any set of points in a drawing. The points could be oil wells, obstructions in a nautical chart, the location of campsites or any other point data.

Step 1: Create a project

Open a new, blank project and import the Europe_eg.mif drawing from MapInfo .mid/.mif format and the Ecities.mfd drawing from Manifold 4.50 .mfd format. Create a map using the Europe_eg drawing, open the map and then drag and drop the Ecities drawing into the map. Format it as shown below.

In this example our project contains two drawings: Europe_eg, a drawing of Europe, and Ecities, a drawing that contains a series of points. A map (called “Cities in Europe” in a stroke of literary inventiveness) shows the two drawings together.

If we click open the map we can see the cities shown as yellow dots on the European background drawing.
The cities drawing by itself is just a drawing that contains points.

If we click open the **Cities Table** table associated with the **Cities** drawing we can see there is one data field, **Place_name**, in the table that gives the name of the city. Note that there are no latitude or longitude fields in the table. [The ID object ID data field that is part of every record associated with an object in a Manifold drawing table has been hidden to keep the table illustrations less cluttered.]

**Step 2: Copy the drawing**

To create a geocoded table we begin by copying the drawing.

Right click on the **Cities** drawing in the project pane and choose **Copy** to copy the drawing.

**Step 3: Paste the drawing as a table**

Next, right click onto any blank location in the project pane and choose **Paste As - Table**.
The Paste As dialog appears to ask which fields we would like to paste into the table for each record. We will scroll down and check the Place_name, [Longitude (I)] and [Latitude (I)] fields. Fields with (I) in their name are intrinsic fields that are automatically computed by Manifold based on the geometry of the objects in the drawing.

There are several options for computed fields that are designed for use when pasting areas or lines as geocoded records. With points we simply choose the [Longitude (I)] and [Latitude (I)] fields.

The result is a new table in the project called Ecities 2 that is not bound to any drawing.

If we click open this table we can see that in addition to the Place_name field it also has Longitude and Latitude fields for each record.

**Step 4: Export the new table**

If desired, we can export the Ecities 2 table to any database format supported by Manifold. To do so we click open the table and then choose File - Export - Table to export it to the desired database format. For example, we could save it as a Microsoft Access database file by exporting it to a .mdb file.

**Notes**
The Manifold CD contains a file called `ecities.mfd` in Manifold 4.50 format that is similar to the `Ecities` file used in this example. The difference is that the version on the CD already has explicit `Latitude` and `Longitude` fields for each city in the drawing's table. If you would like to use the version on CD to follow along with this example, after importing the `Ecities` drawing, open the drawing's table and delete the `Latitude` and `Longitude` columns.

**Important Note when Using 64-bit Manifold Editions**

Due to a lack of required Microsoft facilities in 64-bit Windows systems, Manifold in 64-bit mode cannot export, import, export or link to DB, HTML, MDB, XLS or WKx format files. This includes no access to the MDB parts of Manifold MFD and MapInfo TAB imports. The workaround for importing or exporting such files is to launch Manifold in 32-bit mode by using the Manifold System (32-bit) shortcut, perform the export from or import into a .map project file and then re-launch Manifold in 64-bit mode using the Manifold System (64-bit) shortcut. Linking is more complex: the data must be in some format usable in 64-bit mode within 64-bit Windows systems, such as a SQL Server database. Alternatively, the data can be kept within a Manifold .map project file and linked using the Manifold ODBC driver.
Create a Linked Drawing from a Geocoded Table

In this example we create a linked drawing from a geocoded table created in Microsoft Access. Before beginning this topic, please read the Creating Drawings from Geocoded Tables topic.

Important: If you have installed 64-bit Manifold, do this example in 32-bit mode by launching Manifold System using the Manifold System (32-bit) shortcut. See the note on 64-bit Manifold at the end of this topic.

There are two ways of creating drawings from geocoded tables. A simple, straightforward way is to copy the table and paste it as a drawing. When the drawing is created the coordinates for points in the drawing and their data attributes (if specified for transfer) will be copied from the table. The created drawing (and its table) will be new components that are completely independent from the table that was copied. This process is shown in the Create a Map from a Geocoded Table topic.

Another way of creating a drawing from a geocoded table is to link the drawing from a geocoded table, which may be within the project or which may be some external geocoded table. In this case every time the linked drawing is opened or refreshed it will be dynamically recreated on the fly based on the data in the geocoded table. The position of points and their data attributes will be dynamically taken from the geocoded table. If the geocoded table is altered to add, delete or change the coordinates of records then when the drawing is refreshed the points in the drawing will appear, disappear or move as the table records are updated. This topic shows an example of this process using a geocoded table in an external Access .mdb file.

The distinction between the above two scenarios is subtle, but important. In the first case the data from a table are copied and a new, independent drawing is created. In the second case, a "virtual" drawing is created that remains linked to the table and is controlled by it.

If we have been working through the examples in order we have seen use of import and linking dialogs using simple file types.

In such dialogs the Files of type choice acts as a filter for what is displayed in the browse pane in the File - Import or File - Link dialogs. Simple file types are imported directly from the Import dialog by choosing the desired type in the Files of type box. More complex imports from data sources such as databases are imported by choosing Data Sources ( ) in the Files of type box and then using the Data Source dialog.

Using the File - Link - Drawing dialog takes us into more advanced territory, because whenever we link a file the Data Source dialog is always used to make the connection, even if we would like to link to a simple file type such as Access .mdb.

The reason for this is that a linked drawing is normally a bidirectional, live connection between the linked drawing within Manifold and the originating data source. It is not a simple matter of reading some file, importing the data into Manifold and then forgetting about the original file. Linking a drawing means that Manifold must do all possible to propagate any changes made within the Manifold product back into the originating data source. That normally requires more complicated machinery within Manifold and to maintain modularity within Manifold's internal programmatic structures a more sophisticated access method, the Data Source dialog, is used in all cases of linked drawings.

**Step 1: Create a table in Access**
We begin by launching Microsoft Access and creating a geocoded table called **Towns** in the Access database. The table shows various towns in the United States together with a latitude and longitude location for each town given in decimal degrees. It also includes a URL for the official website for each town and a size field giving the relative population of each town. The fields are all text fields except for the size field which is a number. We save the Access database as a file called **towns.mdb**.

Although Access is a fairly simple DBMS it can be used for multi-user work. More than one program at the same time can use a .mdb database file. In this example we will use the **towns.mdb** database file within Access and at the same time we will use it within a Manifold session by linking to it from Manifold.

**Step 2: Create a linked drawing in Manifold**

Keep the Access session running with the **Towns** table open. Launch Manifold and create a new project. Choose **File - Link - Drawing** to link a drawing into Manifold using a DBMS table.
That launches the Data Source dialog to allow us to choose a data source. This dialog lists previously-created data sources. On the system used for this example there are some spatial DBMS data sources created but not yet one for the file containing the *Towns* table. We’ll create a new data source by pressing the **Add Data Source** button.

![Connect To dialog](image1)

That launches the *Connect To* dialog (very similar to the standard Windows *File - Open* dialog). We choose **MDB Files** in the **Files of type** box and then browse over to the *towns.mdb* file and click **Open**.

![Data Source dialog](image2)

That launches the *Data Source* dialog. We enter *Towns mdb file* for the **Name** and type in the directory path to the *data*townsmdb file and click **OK**.
This creates a new data source in the Data Source dialog, initially named something forgettable like OLE DB Data Source. We immediately rename it to something we will remember, Towns mdb file. We can then double-click the new data source to use it.

This gets us to where we want to go, a Link Drawing dialog. This dialog allows us to choose which table in the .mdb file we want to use to control the drawing and which fields we would like to include in the drawing's table. This is a geocoded table (latitude and longitude values for each record), so the Type is Table with coordinate columns.

We choose the Towns table and choose all columns for import. We use the Longitude and Latitude fields for the coordinates. [Advanced users will be interested to learn that it is possible to choose a query from the database file (if one exists in it) and not just a table. Queries provide a set of records just like a table does and can be used to control a linked drawing just like a table. The only limitation is that parameter queries cannot be used since there is no way to interactively provide a parameter to such queries.]

The dialog also includes a Line ID box that allows us to choose a field that can be used to create lines in the drawing. We will not choose a field since for this example we will only create points. See the Linked Drawings from Geocoded Tables topic for information on creating lines in linked drawings.

Press OK and the result is that four new components appear in the project: a new linked table called Towns, a query called Towns Points and a linked drawing called Towns Points Drawing and the linked drawing's table. Note that the linked components are shown with icons that include a yellow "database" cylinder to indicate they are created from a database.
This is a classic example of a linked drawing created from a geocoded table. Manifold first links a table within the project to the external table, and then creates an intermediate query that transforms the coordinate column data (that is, the data in the latitude and longitude columns) into geometry form so that a linked drawing can be created from that query.

If we open **Towns Points Drawing** we see it contains points, as might be expected from the default name given this drawing by Manifold. In the screenshot shown above we've taken a moment to format the points in a bright green color and to increase their size slightly.

Each point is automatically created from one of the records in the .mdb table, with the data for each point coming from the .mdb table into the linked table in the project, through the intermediate query and then finally into the linked drawing and the linked drawing's table.

If we open the drawing's table we can see that the data attributes for each point are taken from the originating .mdb table as well. All fields appear in this table because we selected all of the .mdb table's columns in the **Link MDB File** dialog. Note that the fields in this table maintain a "live" connection back to the originating external database table. If we change a value in this table the corresponding value in the external .mdb table will change as well.

**Step 3: Show the drawing in a map**

We can import the **US_main** basemap drawing from the Manifold CD, create a map using it and then drag and drop the **Towns** drawing into the map as a layer.
The map shows us that the towns are, in fact, located in the correct geographic locations. To show a more elegant illustration we’ve also taken a moment to specify the projection for the map as an Orthographic projection.

**Step 4: Alter the Access table and see points move**

The useful thing about linked drawings is that if the external database table changes in Access then the linked drawing in Manifold that is created from the table also changes. We can see this by editing the table in Access even as we leave Manifold running.

Using the Windows taskbar we switch back to the Access session.

In Access, let's change the longitude value for Palo Alto.

We'll change it to **-132.15** and then press the pencil icon to make the edit take effect.
The Access table has now been changed. Since Access instantly propagates all changes to the .mdb file in use, the towns.mdb file used to control the linked drawing in Manifold has also changed.

Using the Windows taskbar we switch back to the Manifold session. We click on the map window that displays our drawing and we refresh it to update it with any changes.

In Manifold, we choose View - Refresh Data to make sure that the map in Manifold is updated. When working with Manifold as a desktop application we refresh linked components to see changes. When using Manifold as an Internet map server, the refresh time can be automatically controlled to refresh every given number of minutes or on every image served.

[The Autorefresh View setting seen in the illustration above is used to control windows for things like updating red selection color when items are selected. It does not relate to data refresh as used with linked components.]

After we do a View - Refresh Data command we see the dot for Palo Alto has moved farther West into the Pacific Ocean. As the longitude changed in the .mdb table the point moved to the new location. Any changes made to the external table will appear in the linked drawing as soon as it is refreshed. If a record is added to the external table a new point will appear in the drawing. If a record is deleted from the external table the record's point will be deleted from the drawing.
The creation of linked drawings from external tables is extremely useful, because it allows the use of generic external tables as a data exchange interface for our GIS applications. There are many applications in which the locations of items of interest are captured within a geocoded table. For example, we might have some third party software that monitors the output of GPS transmitters in trucks and updates the location of each truck within a database table, or we might have a table from a web service that publishes the locations of recent earthquakes, the locations of ocean buoys, or the locations of lightning strikes.

There are many more programmers who know how to write code for DBMS manipulation than know GIS, and so as a result there are many more applications that know how to place locations of interest into some sort of geocoded table than know how to interface with any particular GIS package, or, for that matter, with Manifold. The use of linked drawings allows Manifold to interface with such applications without the application needing to know anything about Manifold. The application continues to write locations into the database table as it has long been accustomed to doing. Manifold automatically grabs the data from that table and updates the linked drawing created from it.

In this case, the database system in use is not just a convenient, centralized data warehouse for our organization - it also functions as a "firewall" of sorts between the knowledge needed to get information into the database and the knowledge needed to do something with that information in Manifold.

**Step 5: Thematically format the drawing**

We can use the data fields in the drawing's table just as we would any drawing's table. For example, we can use them to thematically format the linked drawing.

We could format the size of the points and their colors based on the size data field in the drawing's table. As the size field changes in the .mdb file the points will automatically change their size and their colors accordingly. This is a very useful way of presenting varying data in an IMS-driven website.

Note also that double-clicking on a point will launch the hyperlink in the URL field, which is also taken from the table that controls the linked drawing. In sophisticated IMS web sites we can use linked drawings to display information from tables that are changed by other Internet processes. For example, a table might be edited to update the contents of URL fields for each record based on the results of other programs.

**Step 6: Propagating Changes back to the Access Database**

Now, let's do something that usually proves endlessly confusing to beginners. We will edit the longitude and latitude values in the drawing's table and make the corresponding point move in the drawing. But, according to the Tech Tip discussion in the introductory Creating Drawings from Geocoded Tables topic this is not supposed to be possible, true? Let's see for ourselves:
We open the linked drawing’s table and note that, as expected, after we changed the Longitude value for Palo Alto to -132.15 it not only moved the point in the linked drawing but it also changed the corresponding data value in the drawing’s table, which was copied from the Access .mdb table.

We double-click into the cell to edit it...
...and we change the value back to the original value of \(-122.15\).
When we changed the Longitude value for Palo Alto in the linked drawing's table, because of the "live" connection between the linked drawing and its origin, that change was propagated backwards through the linked table back into the Access table, where the value was also changed to 122.15.

![Diagram of cascading links]

We can diagram what happened as shown above. It is not that the change in the longitude cell in the linked drawing's table move through some sort of hard-wired connection from the table directly to change that table's drawing. Instead, what happened is that the change in the linked drawing's table changed the value in the Access table (blue arrow).

When the linked drawing was refreshed, the change in the Access .mdb table was updated into the Towns linked table in the Manifold project and the change in that linked table was propagated through the Towns Points intermediate query into the geometry information used to build the Towns Points Drawing linked drawing.

Linked drawings in this case are different from standalone drawings. Changing the data fields in a standalone drawing's table won't result in any changes in the drawing, since those fields are not linked to anything and are just copies of the data used originally to create the drawing. But with a linked drawing, that linked drawing's table maintains a dynamic link back to the originating table, so changing the fields in the linked drawing's table in a roundabout way comes back to influence the geometric structure of the linked drawing.

This is a bit tricky and sophisticated for beginners, but it is nonetheless a good illustration of how cascading links work within Manifold and how such cascading relationships are preserved even though the data types within each link may be different. We can exploit these capabilities to build IMS sites and other useful applications.

For example, suppose we would like to build an IMS site that allows people to add and move points by editing their latitude and longitude values. A simple way of doing this is to use a linked drawing that takes its data from a geocoded table. Without needing the relatively more sophisticated ability to edit internal drawing geometry we can simply allow users to edit fields in database tables, a task usually well within the skill set of most web developers, who tend to have greater experience with the manipulation of database tables through web sites than they do with the programming details of working with a specific GIS package such as Manifold.

Important Note when Using 64-bit Manifold Editions

Due to a lack of required Microsoft facilities in 64-bit Windows systems, Manifold in 64-bit mode cannot export, import, export or link to DB, HTML, MDB, XLS or WKx format files. This includes no access to the MDB parts of Manifold MFD and MapInfo TAB imports. The workaround for importing or exporting such files is to launch Manifold in 32-bit mode by using the Manifold System (32-bit) shortcut, perform the export from or import into a .map project file and then re-launch Manifold in 64-bit mode using the Manifold System (64-bit) shortcut.

Linking is more complex: the data must be in some format usable in 64-bit mode within 64-bit Windows systems, such as a SQL Server database. Alternatively, the data can be kept within a Manifold .map project file and linked using the Manifold ODBC driver.

See Also

Linked Drawings
Formatting Lines in a Linked Drawing
Formatting Lines in a Linked Drawing

This example shows how to create lines in a linked drawing and then how to format those lines using information stored in an external table. The form of linked drawing used is a limited one normally used only for the display of points from geocoded tables. A more sophisticated form of linked drawings stores line objects in databases uses geometry in tables in a much more powerful way that does not encounter the key issues explored in this example.

However, it is often the case that the only data available to us is not in sophisticated form stored as geometry in databases but instead comes to us in the form of relatively primitive geocoded tables. This topic shows how to use such data. It also embraces the added complexity of leveraging such data to show line objects as an opportunity to teach more advanced notions of how queries can create drawings.

In this example we simulate an application where we have a geocoded table that shows ship tracks, and we have another table in our database that shows information about each ship. We use the two tables together to format the lines that show the ship tracks.

The geocoded table used is similar to the example set of data illustrated in the Linked Drawings from Geocoded Tables topic. Please read that topic before continuing with this example.

Step 1: Create Linked Drawing

We will create a linked drawing from a simple geocoded table. The table is in a comma-separated value format in an ordinary text file called Ship tracks.txt.

If we open the Ship tracks.txt file in notepad we can see it is similar to the tabular data used in the Linked Drawings from Geocoded Tables topic for simple illustrations.

To create a linked drawing, we choose File - Link - Drawing to launch the Data Source dialog and then we add a new data source to the dialog using the same procedure as used in the Create a Linked Drawing from a Geocoded Table example topic. In the Connect To dialog we open the Ship tracks.txt file using a Files of type setting for CSV (comma separated values) text format.
We use the settings shown above in the Link Drawing dialog. Note that the LineID field is used for the Line ID option and the Order field is used for the Coordinate order option.

The result is that seven new components are created in the project pane. The Ship Tracks table is a new linked table that is linked to the external text file. It brings the data from the external file into the Manifold project.

Two new queries, Ship tracks Lines and Ship tracks Points, are also created. Their job is to take the coordinate column information from the linked Ship tracks table and to convert it into geometry form so that it can be used to create drawings. The queries also bring along whatever columns have been requested in the Link CSV File dialog. The Ship tracks Lines query extracts geometry information for lines and the Ship tracks Points query extracts geometry information for points.

Finally, two new linked drawings (and their tables) have also been created. One drawing is created from the geometry information for lines and the other is created from the geometry information for points (Manifold will create a point object for each record in the geocoded database regardless of whether the record is involved in a line or a point).

Open each of the created drawings and confirm its projection by using the Edit - Assign Projection dialog.
**Step 2: Create a Map**

To see the lines drawing together with the points drawing we can create a map.

Pull down the **Create** menu in the project pane toolbar and choose **Map**.

We check both the lines drawing and the points drawing. Since it is usually better style to show points above lines, we will click on the points drawing entry and move it to the top of the stack.
This will show the points drawing as a layer in the map above the lines drawing layer. Press **OK**.

This creates a new map component in the project.

We can click the map open to see that it shows the two drawings as layers with the points drawing above the lines drawing. We confirm the projection of each drawing by clicking on the info bar and clicking **OK** in the **Assign Projection** dialog. After that, we click on the lines drawing layer tab and format the drawing so that lines have a thickness of **3**.
If we open the line drawing’s table we can see that line objects have been created in addition to the point objects, using the LineID field to guide the creation of the line objects. The lines have inherited the LineID values from the points used to create them. Note that the table’s ID column is shaded because it is read-only.

We cannot modify the contents of columns in a linked drawing's table if the drawing is linked from a query and the records returned by the query do not correspond in one-to-one fashion to records in some physical data source, such as a table. In this case, there is not a one-to-one correspondence because the line objects have been created from multiple records in the original table.

Note that a second effect of not having a one-to-one correspondence between line objects and the various points from which they are created is that there is no guaranteed method by default to say what fields or field values, if any, those line objects should inherit from the many points aggregated into each line.

To see why this is the case, consider a line that is created from five points. It is possible that all five points might have exactly the same fields and exactly the same values in those fields. In such a case we may think, “OK, let's have the line created from those points have the same fields and the same values in those fields as well.” But it is not at all guaranteed that such a happy coincidence will occur in all cases. In the case of points representing ship track locations, for example, the time the data for that point was measured, the bearing of the ship at that moment and the sea water temperature and salinity at that location. How should those values, by default, be aggregated into a line object created from such points?

If we were working exclusively with tables within a Manifold project we could have Transfer Rules assigned to say how the "many to one" aggregation should take place, but there are no such things as transfer rules within ordinary text files like the one from which we drew the data, or within plain, ordinary database files of various kinds from which simple geocoding data of this nature is usually sourced. Therefore, the query that is built in such cases to aggregate lines from points simply creates line drawings and can not guess at what is intended.

The ideal, long term solution for aggregating data points into line objects is to use a more sophisticated data storage medium than text file tables. For example, we could use a real database capable of storing geometry in tables such as Oracle spatial or similar, and we could use sensible procedures for building line objects within that database at the time the data is acquired. Drawings could then be linked from such databases and the lines and data attributes for those lines would be created exactly as stored within the data source with no guesswork or manual intervention required.

The pragmatic, short term solution to dealing with such aggregation is to decide what we want out of the data that we have on hand, primitive though it may be, and to make adjustments in the default queries in our Manifold project so that we can have our way with the data as we'd like. This example shows us how.

**Step 3: Create and Link a Table**

For the purpose of this example, let's pretend that these lines represent the tracks of ships. We would like to format the lines based upon an external table.
We launch Microsoft Access and create the **ShipData** table seen above and then save the Access database in a file called **Ship Data.mdb**. There's no particular reason why we used Access to create the table or why we saved it in an **.mdb** file. However, using Access and an **.mdb** file makes the point for the purpose of this example that we can use external programs and external databases to control things within Manifold. We could have, of course, created the table within Manifold and then used **File - Export - Table** to save it to an **.mdb** file, but that would not be as good an example of work outside Manifold.

Using **File - Link - Table** we link a table into our Manifold project using **Ship Data.mdb**. See Importing and Linking Tables for information on how to link a table into a project. If we open the table it appears as shown in the illustration above.

**Step 4: Modify the Query**

So far, so good. We now have a linked drawing that shows points and lines and a linked table that lists some colors and ship names. We need to put the two together using a **Relation** so the colors and ship names appear in the drawing's table.

Before we establish a relation, we need to bring the **LineID** column into the lines drawing. To do that, we modify the **Ship tracks Lines** query to include the **LineID** column.

This is easy to do. We pop open the query, which is shown above. (We've inserted a line break before the **From** statement to allow a more compact screenshot.)
We add a [LineID] column reference as shown by the red arrow above. (The red arrow has been added to the illustration. Manifold does not show red arrows in queries when something new is added.)

We then refresh the drawing by selecting its layer tab and clicking the **View - Refresh Data** command in the main menu. If we open the drawing's table we see that it now has a LineID field brought in from the query from which it is generated, as seen above.

**Step 5: Form a Relation**

Open the drawing's table and launch the Table - Relation command.

Add a new relation that uses the LineID field in both the Ship tracks Lines Drawing Table and the ShipData table as a common field.
Check the **Color** and **ShipName** fields to appear in the relation.

The result is that the drawing’s table will acquire two additional columns, **Color** and **ShipName**, which are taken from the **ShipData** table.

If we pause to think about this for a moment, this is a very interesting situation. This is a relation in a linked drawing, after all, so data in drawing’s table appears in our project as a result of a dynamic linkage to an external table.

We cannot alter the contents of cells in the drawing’s table (which is created by the spatial query that aggregates info from the geocoded table to create lines) but we can add other columns to the table by relations.

The two columns we’ve added are brought in dynamically from a linked table. The linked table appears in our project courtesy of a dynamic linkage to another external table. Everything we see in this table is controlled by two tables that reside outside of Manifold, one table residing in an ordinary text file and the other table residing in an **.mdb** file.

**Step 6: Format the Lines in the Drawing**

Even though the data in the drawing’s table appears dynamically within Manifold as a result of linkages to external tables, we can use the table just as we would any drawing’s table for thematic formatting.
We open the map and click on the Ship tracks Lines Drawing layer to move the focus to the drawing’s layer and then click on the foreground color for lines in the format toolbar and choose Theme to format the lines using a thematic format. We can format the lines by the Color field as seen above.

The result is that the lines are colored red, green, blue or black depending on the value of the Color field.

Again, we have a very interesting situation: the colors of the lines are determined by text fields written into the Ship Data.mdb file that is located outside of Manifold.

**Step 7: Edit the External Table to change Formatting**

Because the colors are taken from an external table, we can change the formatting of by editing the external table.
We can launch Access, open the `Ship Data.mdb` file and change the `Color` value for the first record from `red` to `green`.

Within Manifold, the table that is linked to the `Ship Data.mdb` table also will show a change in the value of `Color` for the first record from `red` to `green`.

If we click on the drawing to give it the focus and then choose `View - Refresh Data` to refresh the drawing's data we see that the ship track in the lower left changes color from red to green.
Refreshing the drawing also refreshes the drawing’s table. We can see that all of the Color values for objects having a LineID value of 1 have changed from red to green.

**Comments**

This example shows how tables outside of Manifold can be used to create points and lines within Manifold and then to format the lines. If we published the above drawing to a Manifold IMS web site, IMS would automatically refresh the drawing for us. We could thus create a web site where other programs could easily write to tables in standard database formats and thus control objects and the appearance of those objects within the maps served by Manifold to the web site.

In this example we had only two data fields, Color and Ship Name, within the external Ship Data table. We did not use the Ship Name field in this example, but it would have been easy enough to create a labels component from this field to show the ship names for each ship track in the drawing.

For that matter, there could be any number of fields in the external table that could be incorporated via a relation within the drawing’s table. We could have fields that provide URLs so that when a line is clicked a hyperlink would be followed to a new web site, perhaps a web site that gives information about a particular ship’s cargo. There are no reasons why external programs could not edit those fields as desired based upon the business logic required for the web site.

We used .txt and .mdb format files for the external tables for no particular reason other than to provide a very simple example that anyone can recreate. While use of .mdb may be fine in a typical web site, in a web site expecting very many visitors it is more likely that the tables used would be hosted on a higher performance database system, such as SQL Server or Oracle.

When recreating this example in an IMS web site, keep in mind the need to make sure the IUSR_login used for the Internet visitor served by IIS has correct access permissions. For example, the IUSR_account must have read permissions for the .map file involved as well as all database files used for linked drawings. If a .mdb file is used, the IUSR_account must also have read and write permissions for the folder in which the .mdb file is located, so that the map server process can create a lock file as is required when working with .mdb.

We used just one field for formatting the color of lines. There’s no reason why other fields could not be used within the external tables to format the points or to format other characteristics such as style or size.

**See Also**

Linked Drawings
Create a Linked Drawing from a Geocoded Table
Storing Drawings in SQL Server

This topic shows how to export drawings into spatial DBMS within SQL Server using Manifold spatial DBMS facilities. We can then link those drawings from the database into our projects and get the benefit of concurrent, multiuser editing. This example requires Enterprise Edition, since storing a drawing into a database using a spatial index for spatial DBMS capability requires Enterprise Edition or higher edition. Once stored in this way, any Manifold edition can import or link the drawing from the database.

The project also shows how to use Administrator Console, a wonderful feature in Database Administrator Edition, to prepare drawings stored on the server for storage of formatting and use of more easily understood friendly names.

This example shows the entire process of uploading a drawing into SQL Server and then linking it into a project. We'll use the SQL Server 2005 Express Edition installation downloaded from Microsoft, so the example assumes we have already installed SQL Server on our computers. Whenever working with SQL Server 2005 (including SQL Server 2005 Express SP 2 or more recent) we should install the Manifold Spatial Extender for SQL Server on the server machine; however, because the Manifold spatial extender is transparent to Manifold functioning (it makes spatial DBMS work with SQL Server 2005 faster and more robust), this example will be identically the same whether we have installed the Manifold spatial extender or not.

The example is written to assume that we are using Manifold on the same machine on which SQL Server has been installed. We can, of course, use a SQL Server installation other than one on the same machine as Manifold and usually will. See the Microsoft documentation for use of OLE DB connections to learn how to set up the dialogs to connect to SQL Server in whatever configuration your installation happens to use.

In Microsoft networks this can be as simple as using the server's name in the connection string instead of (local) for example, if our SQL Server machine's Windows computer name is REDMOND we could connect using REDMOND\SQLEXPRESS instead of using (local)\SQLEXPRESS as cited below.

This assumes, of course, that Windows access permissions and SQL Server have been correctly arranged so that user logins and client machines can connect to and use SQL Server on the server machine. See the SQL Server Express Edition topic for an example of a procedure that enables access to SQL Server from other machines and users in the same domain.

For an example of export to a native spatial DBMS using Oracle, see the Example: Storing a Drawing in Oracle topic.

Step 1: Create a Drawing

The first task is to create a sample drawing we can use to show how to upload drawings in SQL Server. We will do so by importing an existing drawing, since that is significantly faster than drawing a few thousand roads freehand. This also gives us a chance to use this example to show how to import a drawing from the Geocoding Database provided on the Manifold download site.

If we have the optional Geocoding Tools extension installed and have installed the Geocoding Database onto our hard disk, we can import drawings of any county in the United States. To do so, choose File - Import - Drawing and in Import Drawing dialog's Files of type box choose the Geocoding Database Files (states.dat) option.

Navigate to whatever folder holds the Geocoding Database files and double-click on the states.dat file.
The **Import Geocoding Database** dialog opens. Since every Manifold user on the planet is dying to learn more about Carson City, the legendary capital of the state of Nevada, we will use Carson City county in Nevada as the sample county for which we will fetch a drawing. We will also check the **Import Formatting** box so that the drawing appears with default formatting. Press **OK**.

**Note:** Strange as it sounds, there really is a county in Nevada known as “Carson City” even though “City” usually means something explicitly other than “county” in the hierarchical sequence of city-county-state-country. There is such a county name in Nevada because when one has a metropolis of the size of Carson City in the middle of what is otherwise exceptionally bleak, uninhabited terrain the region tends to become known by the name of the sole point of habitation. Carson City is also the state capital of Nevada so no doubt the prestige of such a famous seat of government had a role in the matter of naming the county as well.

Using this particular “county” is slightly misleading within an example topic because some readers will just look at the pictures and not read this text and might therefore believe that this particular importer can fetch drawings for individual cities. That's not the case. It works for county names only and fetches the entire county. It's just that this particular county has a weird, misleading name.

![Carson City NV Drawing](image)

We can open the drawing thus imported and see that it shows the epic scale of Nevada's state capital and surrounding wasteland. Several watercourses are optimistically indicated in blue to show those dry channels in
the desert through which flash floods rage on rare occasions. Contrary to popular belief, it does rain in Nevada, at times strongly enough for water to flow vigorously over the otherwise hard-baked, alkali desert surface.

**Step 2: Save Formatting for Later Use**

We are going to take a moment to save the formatting for this drawing because later on we will need that formatting again.

To do so, click into the foreground color well for lines, choose Theme... and in the Format dialog choose the Save to File command. Save the formatting to a file, perhaps using a name like carson_formatting so it will be easy to remember what that file is for.

**Step 3: Export Drawing to SQL Server**

OK. Enough preliminaries... let's do this. Choose File - Export - Drawing
In the Save as type box we choose Data Sources (). This will launch the Data Source dialog.

The Data Source dialog has not yet had any data sources added to it, so we will click the Add Data Source to add a new data source.

In the resulting Connect To dialog we choose OLE DB Data Sources. Note that we do not use the SQL Server Data Sources option, as that is a special connection used to connect to Microsoft's new SQL Server 2008 spatial (Katmai) spatial DBMS. In this example, we are using Manifold-managed spatial DBMS capabilities within ordinary SQL Server 2005.
This launches the **Data Link Properties** dialog, a standard Microsoft dialog used to connect to OLE DB data sources. In the **Data Link Properties** dialog we choose the **SQL Native Client** and press **Next**.

**Note:** The above assumes we have installed SQL Server Native Client on all client machines so that the SQL Native Client provider is available.
This moves us to the Connection tab, where we enter (local)\SQLEXPRESS for the Data Source. We click the Use Windows NT Integrated security option to log on to the server. This option uses our Windows login status to determine our access to the database. That's usually a more convenient way to set up access to the database (and is the default suggested in the installation topic for SQL Server in this documentation) because most people have trouble enough remembering their Windows login and password let alone a separate user name and password for a database.

Although it is not strictly necessary, we like to press the Test Connection button to see if a connection can be made to SQL Server. It's handy to do that now so that if we made any typographic errors in the Data Source string or if SQL Server is for some strange reason not alive it is best to find that out right now.

When the connection succeeds, a cheerful message box pops open to tell us so. Press OK to close the message box, and then back in the Data Link Properties dialog press OK once more to continue with the export of the table.
Back in the Data Sources dialog we see that a new data source has been added using the default name for the connection. It is usually wise to enter some more memorable name that will help us recognize data sources in a more useful manner. This is easy to do.

In the Name box we can enter the friendly name we would like to use to identify this data source. For this example we will name this data source **Local SQL Server Express**. To use it we either double-click it or we highlight it and press **OK**.
The **Export Drawing** dialog allows us to choose how we will export the drawing, including which columns we would like to export from the drawing's table. Manifold has already helpfully loaded the dialog with reasonable defaults, which we will accept.

We are exporting into a database using Manifold-managed spatial DBMS technology, so the **Type** will be **Manifold**. The pre-built **Geometry** and **Version** columns will be used to convey drawing geometry and make ready for multi-user editing versioning, respectively. The geometry type will be Manifold's native **Geometry** type, the best choice for speed, perfect storage of coordinate system (projection) information and robustness. Manifold will automatically create a spatial index by default.

To store data attributes from the drawing's table we could check or uncheck as many columns as we like. For this example, we will use defaults. We press **OK**.

At this point we have created a drawing and have uploaded it into SQL Server.

**Step 4: Connect to SQL Server with Database Console**

Let us now see how to use within other Manifold projects the drawing uploaded into SQL Server. That drawing will now be available to us whenever we launch Manifold and import or link the drawing from the server.
Suppose it is a few days later and we want to use that drawing. We launch Manifold and choose Tools - Database Console to launch the Database Console.

Database Console remembers the last-used Data source. If we have not previously used Database Console or if we last used some other data source we can instantly get the Local SQL Server Express data source we desire by pressing the [...] browse button to the right of the Data source box in Database Console and choosing Local SQL Server Express in the Data Sources dialog. The Data Sources dialog is extremely useful because once we configure a data source in that dialog it is remembered for future use.

Press the Refresh button to refresh the Database Console display from the Local SQL Server Express data source.
Database console will now light up with a view of our SQL Server database. This particular SQL Server database has been used to store many components. Manifold will helpfully provide drawing, image and table icons to help recognize the different components.

**Note:** If Administrator Console has not yet been used to turn on friendly names in the database the above display is what we see. If we are experimenting with a database in which friendly names have already been established, Database Console will automatically have the **Component View** toolbar button pushed in so that only friendly (that is *component*) names are displayed.

In that case, to see the above display we will have to make sure that the **Component View** toolbar button is pushed out so that we can see all items in the database and not just those to which friendly names have been given.

**Step 5: Link a Drawing from SQL Server**

We will link this drawing into our Manifold project to take a look at what was uploaded into SQL Server. To link the drawing into our project, click on the drawing entry to highlight it and then press the **Link** button in the Database Console’s toolbar.
Manifold knows this drawing was uploaded using Manifold spatial DBMS technology, including a spatial index, so we can choose if we want to import all objects in the drawing or only those within a particular area of interest. This latter option is very useful when working with very large drawings stored in databases. Quite often in enterprise applications we can have immensely large drawings, many gigabytes in size, which cover entire countries or regions as a single, seamless drawing stored within a large database cluster.

However, it is often the case that we might need to work with only a small portion of that large drawing, such as the area of interest involved in some local work, such as the region surrounding a particular city. The area of interest specification provided by this dialog makes such work possible, because otherwise the very large drawing stored on the database server would be far too unwieldy to conveniently use on a desktop machine.

For this example we choose the **Use all objects** option and press **OK**.

A new drawing appears in the project pane, using a database cylinder pictorial element as part of the icon to indicate that it is a linked drawing.

When we click open the drawing we see that it does contain the Carson City drawing, albeit without any formatting. We can only use formatting in such linked drawings if we have previously prepared the database table to store formatting through the use of Administrator Console.
The drawing appears in the correct projection exactly as it was when exported, since Manifold automatically stores full coordinate system information into the database when Manifold spatial DBMS storage is used.

So far in this example we have uploaded a drawing to SQL Server and now we have linked that drawing back into a project. If we are using Enterprise Edition we can have many people work with this linked drawing at the same time doing concurrent, multi-user editing of the drawing, albeit without formatting.

**Step 6: Using Administrator Console**

If we have Database Administrator Edition on hand we can configure the database table using **Administrator Console** so that subsequent use of the drawing by users employing lesser editions is much more convenient. In particular, we can provide a friendly name for the drawing stored and we can turn on formatting.

Assume for the moment that Database Administrator Edition is running on our machine. We launch Manifold and choose **Tools - Administrator Console**. If we have previously connected to the SQL Server database on this machine Manifold will remember the data source used in the **Data Source** box. If we have not previously connected to SQL Server, then we have to press the [...] browse button and use the **Data Source** dialog to connect to the desired data source.

We press the **Refresh** button and the Administrator Console will show the various tables that are in the database, identifying raster data with an image icon. However, because Administrator Console allows more advanced manipulation of tables it will not tag tables with a drawing icon, since we can decide whether a table should be treated as a drawing or not. Just like Database Console, the global filter settings in Administrator Console by default hide various internal tables.

We can determine which table is our uploaded drawing from the name of the table and also by noting the assignment of a projection and spatial index to that table. (The similarly-named table below our drawing's table is the spatial index for that drawing.)

Our first task is to assign a friendly, component name to use for our drawing.
We double-click into the Component cell and enter Carson City as a friendly name.

Our next task is to turn on server-side formatting storage for this drawing. This will allow Manifold to maintain formatting for the drawing within the database. We can do this using either of two methods: we can either open the Properties dialog for the drawing and turn on the Store formatting on server option, or we can turn on the Format column in the main display and click formatting on that way.

To set the properties, we click on the drawing to highlight it and then click the Properties button in the Administrator Console toolbar.

The Properties dialog shows how Manifold is managing this drawing within Manifold’s spatial DBMS infrastructure. As shown above, we check the Store formatting on server box to turn on server-side storage of formatting.
While we are at it, if this drawing will be edited as a linked drawing in a multi-user environment by more than one simultaneous user it pays to check the **Rely on server to increment version automatically** when we are working with a high performance DBMS like SQL Server. This offloads versioning incrementing onto the server for better performance when many users are simultaneously editing objects in the same drawing.

The other way of turning on formatting is to add a **Format** column to the Administrator Console display and to use that. This is just a different way of displaying and setting the same formatting option.

To add the **Format** column to the display, we click the **Columns** button in the Administrator Console toolbar.

In the **Columns** dialog we check the **Format** box and press **OK**.
This adds a **Format** column to the display. We can then double-click into the **Format** cell for our drawing and set the value to **Yes**. This turns on formatting for the drawing.

We can now closeAdministrator Console. Henceforth, any Manifold edition which connects to the database will now be able to use both friendly names and formatting in linked drawings.

**Step 7: Fast linking with Database Console**

The preceding step used Administrator Console to prepare the drawing. Now, whenever someone else connects to that database using Database Console they will have an enhanced experience with that drawing.

Let's assume we connect to the database again using Database Console.

![Database Console](image)

The first thing we see is that the **Component View** toolbar button has been automatically enabled so that the database listing shows only those items for which friendly names (that is, **component names**) have been assigned.

Database Console can tell that friendly, component names (like in a Manifold project) have been designated so it automatically comes up with component names turned on. It also knows to show the **Carson City** item as a drawing.

It is much easier to recognize our drawing when it is shown with the friendly name **Carson City** and when many other database objects do not clutter the display. In the above screenshot, we can see that the particular database we are using has had other Manifold components exported to it, a drawing and an image, and friendly names for those components have been established as well.
We can link the drawing by clicking on it to highlight it and then pressing the **Link** button.

That launches the **Import / Link Options** dialog which allows us to choose to use all objects in the drawing or only those in a given area of interest. The **Version** column is automatically set, as would be the server-side incrementing of version if that had been set using Administrator Console. Press **OK**.

The linked drawing immediately appears in the project pane using the friendly name assigned and a database cylinder in the drawing icon to indicate it is a linked drawing. It has been linked with correct projection information and correct location precision.
We can click open the drawing and see that it is exactly as we expect, albeit without any formatting. It has been enabled to use formatting but, so far, no formatting has yet been specified for the drawing.

**Step 8: Formatting**

Because the drawing has been prepared by Administrator Console to support formatting we can now add formatting. This is easy to do since we thoughtfully saved the desired formatting from the original drawing into a file.

With the drawing open we click into the foreground color well for lines, choose Theme... and in the Format dialog choose the CFCC field and the Load from File command. We load the formatting from the same file to which we saved formatting earlier in this topic.
This reloads the formatting previously saved into a file. Press **OK**.

The drawing now appears with thematic formatting. It has been enabled not only for concurrent, multi-user editing of features, but also for multi-user changes to formatting as well.

**Notes**

The above formatting process is something that is often done by the database administrator at the time the drawing is exported into SQL Server. The usual workflow is to have someone assigned to loading the database with required drawings and to have that person export the table, configure it with Administrator Edition and to assign initial formatting. Because most organizations are naturally protective of their centralized data warehouses and SQL Server installations those functions are normally accomplished by someone trusted to be a database administrator using Database Administrator Edition.
To the beginner the above example from start to finish may seem long because it enumerates every step, including routine steps that experienced users won’t even notice. It’s actually a simple process of uploading a drawing into SQL Server by exporting that drawing to a data source for SQL Server. The first time we create a data source for SQL Server that requires clicking through a few dialogs, but that is done only once. After that, Manifold remembers the data source for us and connecting to it is a simple point and click in the Data Source dialog. The drawing may then be imported or linked from the database using Database Console. Easy!

There are a few extra steps to configuring a drawing in the database for those who want to use friendly names and allow formatting within linked drawings stored in the database. That requires use of Administrator Console and a few extra steps. But the end result is so much simpler for ordinary users if we employ Administrator Console that virtually all organizations working with linked drawings in databases will have at least one copy of Database Administrator Edition available.

This example uses SQL Server Express, a free database product from Microsoft. It’s amazing that we can use free database products in this way, even those which do not have built-in spatial DBMS capability provided by their vendors. That expands usage of spatial DBMS with Manifold to just about every DBMS system one encounters.

One really interesting thing about using linked drawings as in the above example is that we can link drawings from different DBMS sources into the same project. For example, we could link drawings from Oracle, from SQL Server 2005, from SQL Server 2008, from PostgreSQL, from DB2 and even from various alternative DBMS products like MySQL into the same project and then proceed to open those drawings and copy and paste objects from one drawing into another, thus transferring data from within, say, Oracle storage into SQL Server or DB2 storage or vice versa.

Does storing drawings in databases make sense for smaller users, who might not have invested in even a single license of Database Administrator Edition to gain access to Administrator Console, and thus not have the ability to store drawings with formatting? Yes, because even without server-side formatting databases thus used provide vast capacity plus the ability to very efficiently work with only that part of the drawing that is desired. If drawings are not linked but imported, the entire drawing or a portion of the drawing can be imported into the project and then formatted in the usual way, perhaps using a previously saved format as in this example.

For those organizations that wish to perform multi-user editing of formatted drawing "in place," that is, the editing of linked drawings that remain in the database and are not imported into a Manifold project, the cost of a single license of Database Administrator Edition is still a bargain. Only a single license is required (to upload drawings and to enable formatting with Administrator Console), since all other users can import or link drawings from Manifold spatial DBMS using any Manifold edition, even Personal Edition, and from native spatial DBMS (such as Oracle, DB2, SQL Server 2008 or PostgreSQL) with Enterprise Edition.

A further reason to use spatial DBMS for storage is that only Enterprise Edition is required to upload images, which any Manifold edition can then import or link into a Manifold project. Linked images stored in databases are amazingly fast, as fast as using a linked image from an ECW. A very active GIS organization or professional user will therefore almost certainly will already have a database server in play if for no other reason than to store linked images in a high performance way.

**Technical Note on Primary Keys**

This example uses SQL Server, which has full support for the use of primary keys; however, even though it seems that all of the mainstream databases allow creating a primary key for a table, it is possible that one might encounter some database that does not. When exporting a drawing into a spatial database, Manifold attempts to create a primary key column in the destination database table using the OID field cited in the dialog at the end of the third step above.

Having a primary key column is important, because many ODBC drivers and OLE DB providers expose limited functionality for tables that do not contain a primary key. Depending on the software in use, a primary key can be one key column or a combination of columns, which taken together form a unique key.

If a table does not contain a primary key, some drivers will make that table read-only while some other drivers will consider that table read-write but with restrictions. A typical restriction when a primary key is lacking might be for the driver to make the table read-write so long as the database cursor (an internal programming concept) is forward only, which means that the client software can access row A and row B and row C in forward order but cannot access row A, then row B and then row A again.

Manifold contains a lot of internal code to deal with such goofy limitations and tries mightily to work around the limitations of various ODBC and OLE DB drivers. Manifold can even make some tables with restrictions like those above read-write in some cases. But no matter how Manifold may strive to work around such limitations arising from lack of primary keys it cannot do so in all cases.
A further reason for using primary keys is that many drivers contain optimizations aimed specifically at tables with primary keys. As a result, the driver and any client using that driver, such as Manifold, will perform much faster on tables containing primary keys.

The above limitations of drivers may also affect the ability of Manifold to interact with tables without a primary key in matters other than simple access. In some cases, linking a drawing from a table without a primary key might disable editing objects, but still allow adding or removing objects. In more extreme cases, the drawing might end up being read-only.

For these reasons, it is important to always use a DBMS that allows creation of primary keys and also not to later manually go into the database and mess with the primary keys Manifold creates. Manifold can work with a very wide range of DBMS packages that use a seemingly endless variety of ODBC and OLE DB drivers, so this is not a problem for anything remotely resembling normal use. However, programmers bent on doing custom, tricky things with the innards of databases should consider themselves forewarned.

**Tech Tip: Deletions in Database Console**

Deleting an unwanted drawing stored to SQL Server can be complex. See the discussion in the Database Console topic.

**See Also**

Spatial DBMS  
Manifold Spatial DBMS Facilities  
Database Administrator Edition  
Database Object Projection  
Database Object Properties  
Geocoding Tools  
Geometry in Tables  
Import Drawing - Geocoding Database  
Linked Drawings  
Multi-User Editing of Linked Drawings  
SQL Server Express Edition  
Tools - Administrator Console  
Tools - Database Console  
Using Administrator Console
A Flashy Demo - Web Queries and KML

Every Manifold advocate should have at least one flashy demo that he or she uses to show off Manifold. While it is the deep power of Manifold doing things like topology overlays that impresses the GIS elite, to avoid putting to sleep those guests who may not have a deep appreciation of GIS it is important to keep the demo light and visually appealing.

Important: If you have installed 64-bit Manifold, do this example in 32-bit mode by launching Manifold System using the Manifold System (32-bit) shortcut. See the note on 64-bit Manifold at the end of this topic.

This examples shows a sequence of moves using Excel to grab dynamic data from the web, using it in Manifold and then grabbing part of that data and exporting to Google Earth. Fast and easy, it's a proven crowd pleaser.

Practice this routine. It's slow at first but after a few practice sessions can be accomplished with confidence in less than ten minutes. It goes so fast you'll spend more time explaining various wonderful things about Manifold than doing the demo.

This demo requires a web connection for the machine being used. In addition to Manifold we also need Microsoft Excel and Google Earth installed on the machine.

Step 1: Create a Web Query in Excel

Excel can be setup to automatically grab data from a web site using an Excel facility called web queries. Experts can create a web query in Excel by opening an Internet browser to the web site desired and then dragging and dropping the target data into an Excel session, using a special Paste As option. That's very flashy but easy to do wrong.

Another way to create a web query in Excel is the more traditional way using Excel menu commands shown in this topic. This is easier to remember, still flashy and virtually impossible to get wrong. For this example we will use the traditional way to create a web query that grabs data about recent earthquakes. Here's how:

Launch Excel
Choose Data - Import External Data - New Web Query

The resulting New Web Query dialog launches like an Internet Browser. It opens to your home page so you might want to set your home page to something respectable before doing this for an audience.

In the Address box, enter the URL to the USGS web page that gives a list of recent earthquakes in the US greater than magnitude one. The URL happens to be...


...but that's not possible to remember. Most people will launch an Internet browser just ahead of the demo and then use their favorite search engine to search for recent US earthquakes USGS or some similar search terms. That will find the USGS earthquake page, which has links to pages giving lists of recent earthquakes in the US and worldwide.

In fact, some people like to start this demonstration with a web browser already on the web site showing the list of recent earthquakes saying something like, "Wouldn't it be great if we could show the data in this page on a map in our GIS system and have it automatically updated, all without any programming?" A good performer always dangles a bit of bait in front of his or her audience to get their attention.

Once you have the page in your Internet browser you can copy and paste it into the Address box of the New Web Query dialog and press the Go button.
The earthquakes page can take tens of seconds to appear, because the USGS server is generating a table of recent quakes and sending it out to the page. This is your chance to distract the audience from the delay by talking about how this technique will work with any web page of locations, such as navigation buoy locations, locations of storms, locations of tracked animals or vehicles and so on.

When the page appears, the **New Web Query** dialog will place a yellow box with an arrow at the location of every table that is embedded in the page.

To find the data of interest, scroll down on the page (scroll bars in the dialog are not visible in the screenshots shown) to get to the table of recent earthquakes. The table is marked with a yellow box with an arrow. Note that the table has the magnitude, time and latitude and longitude locations of the quakes. Those who have been following along in previous examples know it is a **geocoded table**.

Press the yellow box with an arrow to select the table.
When the table is selected it will be highlighted and the box next to it will turn green.

Press the **Import** button on the **New Web Query** dialog.

That launches the **Import Data** dialog. Press the **Properties** button.
We are here in the **External Data Range Properties** dialog mainly for one very interesting box, the one we just checked that says **Refresh every ... minutes**. We’ve set the value to 30 minutes. What is immensely cool about this box is that we can tell Excel to go fetch data in background from the website so it always has current data in the spreadsheet. Our web query is like a robot in Excel that interrogates the USGS list of recent quakes every 30 minutes to get the most recent list.

In this case, we’ve told Excel to grab a table of recent earthquakes and to update the data from the web site every 30 minutes. That seems often enough for quakes. Press **OK** to exit the properties dialog and then **OK** to exit the **Import Data** dialog.

Back in Excel we see that the spreadsheet now contains data from the web site. If we were to sit around for 30 minutes we could see the numbers change as the spreadsheet automatically updates itself from the web query.
The table looks a bit messy, with missing values and a first column that looks like junk, but Manifold will digest this nonetheless.

Save the Excel spreadsheet to a file called **usquakes.xls** in some convenient folder.

**Step 2: Create a Linked Drawing in Manifold**

Launch Manifold and choose **File - Link - Drawing**. This immediately launches the **Data Source** dialog to allow choosing a data source from which the drawing should be linked.

We click the **Add Data Source** button to add a data source for our desired **.xls** file.

In the **Connect To** dialog choose **XLS Files** in the **Files of type** box and navigate over to the **usquakes.xls** file and open it.
This adds a new data source, but the default name used is not particularly helpful. We will give it a more memorable name.

In the **Name** box we enter a name that will help us remember what this data source is for. Press **OK**.
In the resulting Link Drawing dialog Manifold has already examined the .xls file and has pre-loaded the dialog with sensible choices. Manifold knows that LON likely means the longitude field and LAT the latitude field so these come pre-loaded with the right field choices. We’ll select all the fields except the first F1 column which is just empty junk at the beginning of the Excel table.

Pausing a moment to enjoy the priceless value of a demo where most dialogs work fine using default values, we press OK.

The result is that an ensemble of new components appear in the project. When giving this demo to an audience of very sophisticated GIS people who understand something about spatial SQL, we might take a few minutes to discuss how Manifold works with geocoded tables, using the information in topics like the Creating Drawings from Geocoded Tables topic. However, tread lightly on such matters when beginners are in the audience.

The new components created do have ugly names because of the naming conventions within spreadsheets, but this is an opportunity to do something with Manifold that even a novice will enjoy: we will rename a file.
Right click onto the linked drawing and choose Rename. Rename it to US Quakes so the audience can see it automatically renames dependent components and that doing things in Manifold is really easy, using clicks and context menus as one would expect in Windows.

When we click open the drawing the GIS people are impressed to see a drawing created on the fly from geocoded table in the XLS, but less technical people may be bored by the lack of pretty colors or lack of a map. Let's add pretty colors first and then later on (whet their appetites) we'll add a map.

**Step 3: Format the Drawing**
Click into the background color well for points in the formatting toolbar and choose Theme.... In the Format dialog use the MAG field with Natural Breaks and the Spectrum palette. We like to change the first color in the palette to a darker purple. Doing so shows how easy the dialog is to operate. Press OK.

After doing this demo once or twice most people cannot resist showing off different palettes while clicking the Preview box on and off so that people can see how interactive Manifold is. Cheap and flashy, perhaps, but fun and impressive as a matter of detail.
Examples

The result shows earthquake points colored by their magnitude, with more dangerous quakes colored more towards the red end of the spectrum. Someone in the audience will almost inevitably ask if points could be made larger for more powerful quakes.

Click into the size well for points and click **Theme...**. In the **Format** dialog choose a size range for points similar to that shown above. Sometimes it is fun to turn on the **Preview** box and ask for audience feedback as different sizes are tried. Getting the audience involved is Rule Number One for performers.
That's better. Now we see larger quakes with larger points. By now the audience is clamoring to see the points on a map. They've probably asked for this before but have been put off by the demonstrator. Making the audience ask for more is Rule Number Two for a performer.

**Step 4: Create a Map**

![Create Map dialog](image)

Import the **World_eg** sample drawing from the Manifold download site. The drawing is shown above, but the steps to import it are not shown in this example as the elementary importing of drawings is covered in many other places.

Right click on the **World_eg Drawing** in the project pane and choose **Create - Map**. In the **Create Map** dialog check both drawing boxes to participate in the map and press **OK**.
Open the map to oohs and aahs as people see where the quakes are on a real map. Pan and zoom a bit. Note that using the USGS site ends up creating at least one point at the 0,0 origin of latitude and longitude (by Africa) because the first row in the table after the column names does not have lat/lon values. You can promise the audience that later you’ll show them an expert Manifold trick that can be used to whip such deficient data into shape. For now, we ignore it.

**Step 5: Create a KML File**

Everyone likes to share the fruits of their work, so you can set up the next step by telling the audience you’ll show them a fast and easy way of communicating the data you have to others. We can do that by creating a PDF (a variation of this demo) from a print layout, or use the following technique of creating a Google Earth KML file that can be emailed to colleagues.

**Note:** Drawings ideally should be in *Latitude / Longitude* projection before export to KML or KMZ. Our drawing already is in *Latitude / Longitude* projection. See the note at the end of this topic for discussion of projections and KML.

Zoom into Southern California where there are always plenty of small earthquakes and select some of them (make sure you are on the quakes layer in the map).
With the quakes selected, choose **Edit - Copy** to copy those points.

Right click onto a blank spot in the project pane and choose **Paste**. A new drawing appears, helpfully named in a like fashion to its parent. GIS people in the audience with Windows skills know they are seeing something very interesting here, as this sort of thing is not done with legacy GIS systems.

Open the drawing and people will see that the formatting is retained. In the above illustration we've taken one step more (not shown) and have changed the thematic formatting for point size to increase the point sizes a value or two, say, from 2 to 4 and so on, so that the points appear larger. We have a reason for doing that, as will soon be clear. We've also renamed the drawing to **SoCalQuakes**.
With the SoCalQuakes drawing open, we choose File - Export - Drawing. In the Export Drawing dialog we choose KML / KMZ Files in the Save as type box and we provide the filename socalquakes.kml as the File name. Press Save.

The Export KML File dialog pops open to ask us which fields we would like to use for Name and Description. We'll use the MAG field giving magnitudes for the Name and the date and time field for the Description. Press OK.

What we've now created is a file called socalquakes.kml that is a "document" file for Google Earth. We can email it to someone and if they have Google Earth installed on their machine when they double-click the socalquakes.kml to open it, Google Earth will launch and display the contents of the file.

**Step 6: Display KML File in Google Earth**

Switch out of Manifold and into Windows Explorer and browse to the folder where the socalquakes.kml was stored. Double-click the socalquakes.kml file to display it in Google Earth. Google Earth will launch automatically and will zoom in from space to show our earthquake points in southern California.
The MAG field we used for Name will result in the points in Google Earth being labeled with the magnitude of the earthquake represented by each point. The points are still colored by the strength of the quake because that's how they were sent out by Manifold. The points are large and obvious against the clutter of the photographic background because we enlarged point size slightly before exporting the drawing.

If we look at the Google Earth information panes we can see that individual points have a description of sorts which is the date and time field. This is not the most powerful interface for handling data information, but then Google Earth is about visual interest, and lots of it, and not about manipulating or displaying data.

**Step 7: Adult Talk about Spatial Queries**

Google Earth is fun for all ages, but some folks with GIS expertise might be interested in just how you propose to use spatial magic to get rid of that spurious point near Africa. That's a chance to talk about how the drawing created from the XLS was actually created by an intermediate query from the linked table in the drawing.

```sql
OPTIONS CoordSys("Latitude / Longitude");
SELECT [Sheet1$].\$,
    AssignCoordSys(NewPoint(CAST([LON] AS LONGITUDE), CAST([LAT] AS DOUBLE))
FROM [Sheet1$];
```

You can open the query so that people with some familiarity with SQL can see how it works.

```sql
OPTIONS CoordSys("Latitude / Longitude");
SELECT [Sheet1$].\$,
    AssignCoordSys(NewPoint(CAST([LON] AS LONGITUDE), CAST([LAT] AS DOUBLE))
FROM [Sheet1$] WHERE [MAG] > 0;
```
The next step is to add just one line at the bottom, `WHERE [MAG] > 0`, which adds a qualifier to do all that spatial query stuff only for cases where the magnitude of a quake is greater than zero. Spurious lines in the table that don’t have data for latitude and longitude won’t have earthquake magnitudes greater than zero either, so this is a quick and dirty way of accomplishing the task.

After modifying the query in the above manner you’ll have to Run the query and then Relink the drawing back to the query. Practice this a few times before doing a live demo so you feel confident with the operation and show that confidence in your demo.

Operating the Relink dialog shows people how the drawing is linked to the query, and GIS people with computer science expertise are never failed to be impressed how the system can hang together so well that after the Relink the formatting is still intact.

The result is a map of earthquakes with no spurious points near Africa.

**Projections and KML Format**

Google KML and KMZ formats support Latitude / Longitude projection only. Drawings to be exported to KML or KMZ can be in any projection and will be re-projected on the fly into Latitude / Longitude during export; however, to avoid a potentially slow re-projection process it is wise to explicitly re-project drawings into Latitude / Longitude before export. This not only avoids a slow export, it also provides an opportunity to catch errors before export if the re-projection results in a strange display as might happen if the initial projection was not correctly assigned, say, after importing projected data from some format that does not store projection information.

Unlike drawings, images must be already in Latitude / Longitude projection to be exported to KML or KMZ.

**Notes**

Some demonstrators like this demo so much they go on to create a web site in Manifold IMS. this is especially interesting because as Excel refreshes itself with recent earthquake data Manifold IMS can also be told to refresh itself at a specified interval. This is a fast and easy way of creating a web site showing current earthquake data, with less than 15 minutes required for the entire process. If you add IMS to the demo, practice creating the web site on the target machine so that something dumb, like forgetting a Windows permission, won't trip you up in the live demo.

In addition to using export to Google Earth, many demonstrators will create a linked image in the Manifold map, taking imagery from TerraServer or other image server. That shows how Manifold can be used to bring together data from different web sources within the same project. For example, some people bring the World_eg
background into the map as an image linked from a Manifold Image Server. If you use that idea in a demo, work it out in advance so that you create a map from the linked drawing (which cannot be re-projected) and then show earthquake points on that.

Once pasted, the points copied from southern California will create a static drawing that has no link back to the original XLS. That's OK, as it illustrates how Manifold can grab data and move it from remote to local form.

This demo example was inspired by the magnificent presentations given by Manifold expert Dr. Arthur Lembo, who has used web queries together with Manifold at Cornell University to create rapid response web sites to help with disaster relief.

**Important Note when Using 64-bit Manifold Editions**

Due to a lack of required Microsoft facilities in 64-bit Windows systems, Manifold in 64-bit mode cannot export, import, export or link to DB, HTML, MDB, XLS or WKx format files. This includes no access to the MDB parts of Manifold MFD and MapInfo TAB imports. The workaround for importing or exporting such files is to launch Manifold in 32-bit mode by using the Manifold System (32-bit) shortcut, perform the export from or import into a .map project file and then re-launch Manifold in 64-bit mode using the Manifold System (64-bit) shortcut.

Linking is more complex: the data must be in some format usable in 64-bit mode within 64-bit Windows systems, such as a SQL Server database. Alternatively, the data can be kept within a Manifold .map project file and linked using the Manifold ODBC driver.

Although Microsoft is slowly moving forward towards 64-bit operation of Office and the JET components used by Access and Excel, at the present writing (2010) this is such a total kludge that while it is possible to get 64-bit access to Access or Excel it is just easier for most to continue using the 32-bit versions. See comments in the 32-bit and 64-bit Manifold Editions topic.

**See Also**

Linked Drawings
- [A Flashy Demo - Web Queries and KML](#)
- [Exporting KML to Google Earth](#)
- Fun with Google Earth
- [Linked Images from Google Servers](#)
- Export Drawing - KML, KMZ
- Export Image - KML, KMZ
- Import Drawing - KML, KMZ
Creating Bordered Lines

When illustrating roads we will often want to show principal highways or other roads using lines that have "borders" on the edge of the lines with the borders naturally being spliced together at intersections.

This is an easy effect to achieve in Manifold. Although Manifold includes a bordered line style, using the default bordered line style does not achieve "splices" at places where several line segments come together. The technique described in this topic will create seamless splices at such intersections.

Step 1: Import and clean up the drawing

For this example we will import the `bay_roads_eg.mif` drawing.

After importing, we will create a map from the drawing, open the map and zoom in as seen above. Note that the example drawing contains points at the intersections of the road lines. We will eliminate them.

To do so, we set selection modes to select points and to not select areas or lines.
Choosing **Edit - Select By Type** will select all of the points in the drawing.

We can then choose **Edit - Delete** (or press the Delete button or press the Delete key) to delete the points.

**Step 2: Duplicate the layer**

Next, we will duplicate the layer. To do so, right click on the **Bay_roads_eg Drawing** layer tab and choose **Duplicate**.
A new drawing layer called **Bay_roads_eg Drawing 2** will appear in the map just above the original layer. The map will not change appearance since the duplicate exactly overlays the original layer and (thus far) is formatted identically.

**Step 3: Format the layers**

The final step is to format the two layers to provide the desired appearance. We begin by clicking on the lower layer tab (**Bay_roads_eg Drawing**) so that the focus is on that layer.

![Image of map with layers](image)

Change the thickness of the lines in that layer by clicking on the size button for lines in the Format toolbar and choosing 6 for the size.

![Image of map with thicker lines](image)

The result is that the lines in the lower layer become very much thicker.

We now click on the layer tab for the upper layer (**Bay_roads_eg Drawing 2**) so that the focus is on that layer.
Change the color of the lines in that layer by clicking on the foreground color well and changing the color to yellow.

With the color change we can see the yellow lines in the upper layer above the thick black lines in the lower layer.

We can now increase the thickness of the yellow lines in the upper layer by changing their size to 4.

This provides the desired effect.

Variations

We can create many variations by mixing different format styles in different layers.
For example, if we format the upper **Bay_roads_eg Drawing 2** layer as seen above we can create dotted line highways.

The dotted lines in the center of the highways are created because the upper layer uses black for the dots and a wider, size 4 yellow line for the outline of the dotted line.

We can create maps of great complexity using the above method. In fact, we can combine many layers into a map and arrange Zoom Ranges on the layers so that as we zoom in road lines that are initially shown as single, thin lines begin appearing as bordered lines as seen above. To do this, specify **Zoom Ranges** so that the layers showing thin lines switch off at the zoom level at which the bordered line layers switch on.

**Tech Tip: Interweaving Lines**
Maps of great depth can be created by stacking layers and using the formatting shown above. However, all objects in a given layer will be seen above all objects in layers below them. On occasion we may want to create an "interwoven" line that is sometimes above and sometimes below objects in different layers. This requires additional work, but it can be done if the visual effect merits the extra work. The rest of this topic describes an intermediate method that would rarely be done but is nonetheless interesting as an example of the great flexibility of Manifold tools.

Suppose we wish to draw an interwoven line that is sometimes above the yellow roads and sometimes below them. We create two drawing layers called above and below where the above layer is above the yellow roads layers and the below layer is below the yellow roads layers. Beginning at the lower left of the curved black line we draw a line in the above layer. We then click on the below layer to move the focus there. Turning on Snap to Lines, we can then draw another line that continues the curve, using Snap to be sure it starts exactly at the end of the previous line. We finish this next line before we need to go above the yellow roads.

We then click on the above layer and draw a line that continues the curve above the yellow roads. We continue in this way switching between the above and below layers as necessary to draw the line. The illustration above has had the above layer formatted in red and the below layer formatted in blue so we can see how the intertwining line is made up of many different lines in the above and below layers.
To achieve a bordered line effect for the interwoven line we duplicate both the above and below layers. The lower layer of each pair we format in black color with a size of 6. The upper layer of each pair we format in bright green with a thickness of 4.

An unpleasant effect occurs at the end of each green line where a black border appears. Because Manifold rounds the ends of lines with increased sizes, the black pixels at the end of the above layer are appearing above the green pixels of the Below 2 layer. Curing this is tedious, but possible.

The visually best way of fixing this is to create an additional drawing layer called ends and creating a line that crosses the boundary between the green lines.
If we zoom in we can see how these lines were created. With Snap to Lines turned on we can create a line that runs from the next to last coordinate of one line to its end to the next to last coordinate of the following line.

Zoomed out once more, we format the ends layer in green color with a size of 4. This achieves a seamless effect of an intertwining green line that sometimes goes above and sometimes below the yellow lines. Because the pixels line up perfectly there is no visual clue that the effect is made up of many layers.

The above exercise could be very tedious for a large map. Is there an automated way to sometimes move a line above and sometimes below another layer? If our data set consists of multiple lines where lines at intersections have some data attribute code such as a bridge or underpass code we can select "above" segments by this attribute, cut them out of the initial drawing and paste them into an "above" drawing layer.

If our drawing has no such data attributes, we can take a semi-automated approach of using a grid of lines at suitable resolution and the Intersect Lines transform to chop up a continuous line into many smaller lines. We can then visually select all portions that should be "above" the other lines using Add to Selection mode, cut them out of the layer and paste them into an "above" drawing layer. We can then use Join Lines to reassemble into a smaller set of lines the many lines created by the Intersect transform and left behind in the original layer.

In both of the above cases, creating "end" lines to bridge the dark pixels that appear between lines will be tedious, but at least we will not have to create the many parts of the intertwining line itself.
Georegister a Scanned Paper Map

A common use of Manifold is to create drawings that are based on paper maps. Although many maps are already available as drawings in digital GIS formats there are still many maps published on paper in the world's archives. To use information from paper maps we first use a scanner to create an image. We can then import the image, georegister it, and use it within Manifold.

This example uses a map originally published in a book that shows climatic zones in Australia. Our objective is to get this image into Manifold and to georegister it. The image is a small one printed as a small part of a page in an atlas. However, the data is interesting and we would like to make as much use of it as we can.

Before reading further in this example please read:

- Georegistration
- Georegistering an Image to Known Coordinates
- Error Surfaces
- Create a Table and Add Records

This is a long example even though the material it covers is quite elementary. It is lengthy because it includes many small details that a reasonably experienced Manifold user would take for granted.

In most cases of georegistration we will georegister an image to a drawing where both image and drawing contain discernable geographic features that can be used as control points. For example, the image might clearly show an intersection between highways that is also obvious in the drawing. Georegistration in such cases is fairly obvious and how to do georegistration in those cases should be readily apparent based on the topics cited above.

This example takes a much more complex case, which goes into a major diversion to create "control points" within a table. It shows how even if we don't have a drawing to which we can georegister we can still georegister if there are locations within the image that can be assigned latitude and longitude locations. In this example those locations are intersections of graticule lines in a printed map. In other cases, the locations within the image might be some clearing or other visible location for which we have latitude and longitude locations, perhaps because we have visited such locations with a handheld GPS and have recorded their coordinates.

For example, it's often the case in forestry applications where an image of a forest must be georegistered but there is no drawing to which it can be georegistered. In such cases foresters might visit actual locations seen in the image and note the latitude / longitude coordinates of such locations using a handheld GPS. Such locations might be clearings, intersections of logging roads or other features that can be distinctly identified in the image.

For this example we assume the Project pane and the Control Points pane are both open.

**Step 1: Scan the Image and Import into Manifold**

We lay the book on our scanner and scan in an image. Our scanner produces a .tif or other graphics format image that we can import into Manifold. We crop the image to Australia and convert it to RGB. We've named the image Oz, a friendly nickname for the enchanted land of Australia.
The paper map was printed in the book using a Winkels projection. When imported into Manifold the image is imported using the default Orthographic projection. We really don’t care what projection was used initially or how it was imported since the georegistration process will re-project the image as needed. Likewise, we really don’t care if the book was placed on the scanner with imperfect alignment. If the image was rotated the georegistration will remedy that as well.

**Step 2: Place Control Points in the Image**

Most paper maps have a graticule (lines of latitude and longitude) printed on the map. The graticule lines are usually placed at round numbered latitudes and longitudes.

If we examine the book we see that the graticule lines for our example have been placed every 10 degrees of latitude and every 15 degrees of longitude. By inspecting the map we can immediately read the latitude and longitude location of each graticule intersection. For example, in the illustration above the intersection immediately to the North of the printed "Wellington" caption is located at longitude 180 and latitude -40. [South latitudes and West longitudes are written as negative numbers.]

We will use graticule intersections in the image as control points. We intend to follow a procedure similar to that given in the Georegistering an Image to Known Coordinates topic.
Opening the Control Points pane we click on the **New Control Point** button and add control points at graticule intersections near Australia.

We name each control point with a short name that helps us keep track of which control point is which. When placing control points in a regular grid it is often helpful to use names like A1, A2, A3 and so on for the control points in the first row and then B1, B2, B3 and so on for the next row.

Note that when placing control points on graticule lines in this way all of the A points have the same latitude value, as do all of the B, C and D points. Likewise all the points on the same longitude line have the same longitude value. We will exploit this coincidence later in this topic to enter more rapidly the values of control points that will be used to guide georegistration.

We do not show the full illustration in this example, but the printed book shows that all of the A control points are at latitude -10, the B points are at latitude -20, the C points are at latitude -30 and the D points are at latitude -40. The 1, 2, 3 and 4 points are mostly at longitudes 105, 120, 135 and 150. The D4 control point is at longitude -165.

**Note:** When placing control points, it is often helpful in Tools - Options to uncheck **Autoscroll window on edit or selection process.** This prevents the image window from scrolling as we move the control point insertion cursor into the window. Also, it is often helpful to zoom into the image for each control point so we can place it more precisely.
After we finish placing control points the control pane will have a list of control points as seen above. [The coordinates for each don't really matter since they are abstract coordinates giving their position within the Orthographic coordinate system of the image.]

**Step 3: Create a Table with Control Points**

By reading the paper map we can determine the exact coordinates of each of the sixteen control points marked on the image. This will allow us to use a procedure like that in the Georegistering an Image to Known Coordinates topic. In that procedure we created a blank drawing and placed control points in the drawing and specified the latitude and longitude coordinate of each control point in the Control Points pane.

In this procedure we will take a slightly different approach. We will first create a table and then enter control point values into that table. We will then copy the table and paste it as a drawing of points. We will then load the control points pane using those points. This is a more indirect approach, but when large numbers of control points must be entered it is usually faster in the long run to exploit the editing and organizational powers of Manifold tables.

Follow the procedure set forth in Create a Table and Add Records to create a new table and to add sixteen records to it. Create each record with a value in the Name field only. Because this example places control points at the intersection of round-numbered latitude and longitude lines we will enter whole numbers only. It's often the case that we will want to enter latitudes and longitudes using either decimal degrees notation or degrees - minutes - seconds notation. Manifold tables can accept entry in either style as shown in the Create a Table and Add Records topic.
For each record we will specify only the Name, using names corresponding to the names we used for control points in the image. Since we don't specify the latitude or longitude value, each record will be created with a 0 value for latitude and longitude. The table appears as seen above after we have added sixteen records to it.

Now, a cool Manifold technique: select all of the A records. Double click into the Latitude cell of one of the selected records and enter -10. [For a refresher on selection techniques in tables, see the Selection in Tables topic.]
When we press Enter the value -10 will be placed into the Latitude cells of all the selected records. This is a fast way of entering the same value into more than one record. If we look back at the illustration showing control points in the image we can see that all of the A control points are on the latitude -10 graticule line.

We continue in this way to add latitude values of -20 for the B records, -30 for the C records and -40 for the D records.

We now will add Longitude values. Note that A1, B1 and C1 all have the same longitude. We select these records and double click into the Longitude cell for one of the selected records and enter 105.
When we press Enter the 105 value appears in the Longitude cell for all of the selected records. We next will select the A2, B2, C2 and D1 records and enter 120 for the Longitude of those records. [At this point we notice that it would have been smarter to number our D records as D2, D3, D4 and D5. Had we had the foresight to do so then all of the "1" names would have the same longitude, as would all of the "2" names and so forth.]

This trick of selecting several records at once to enter the same value in all of them is a very useful technique. If we don't want to use it we could always simply enter the latitude and longitude values for each record one at a time. However, taking some time to think through our task and planning to take advantage of patterns in our work allows us to work more rapidly and with less chance of error. It's a good example of how planning combined with skill in using Manifold results in more efficient workflow.

The result of our work is a table that contains records giving the names and locations of control points.

**Step 4: Create a Drawing from the Table**

To create a drawing from the table we Copy the table in the project pane and then Paste As a drawing.
In the Paste As dialog we choose only the Name as a field to be pasted as a data attribute into the drawing. The latitude and longitude of each point is implicit in the point’s position so it is not necessary to paste these. Because the point locations were entered into the table as latitude and longitude degree values, we check the Latitude / longitude coordinates box (the default setting).

See the Create a Map from a Geocoded Table topic for another example of creating drawings from tables using Copy and Paste As.

The Paste As operation creates a new drawing in the project. We rename it to My Control Points Drawing and open it. The drawing shows sixteen points that were created from the records in our table. It is in Latitude / Longitude projection. We will now use these points as control points.

To do so, we go to the Control Points pane and click the Load Points button. This dialog will create control points from all of the points in the drawing, using the specified field as the name of the control points. In the Load Control Points dialog we will use the Name field (of course) as the name for the control points. Press OK.
The result is that control points appear in the drawing at the location of each point. Each control point is named using the Name field of the point at which it appears.

**Step 5: Georegister the Image**

We can now georegister the Oz image using the control points specified in the My Control Points Drawing drawing. To do so we click on the image to make it the active window.

Press the Register button in the Control Points pane to launch the Register dialog.

We will use My Control Points Drawing as the reference. With sixteen control points in what is intrinsically a low accuracy project (we did, after all, scan a small image in a printed book) the Numeric method with order 3 will provide reasonable accuracy. We will also check the Save error surface as box to create an error surface so we can later see what the likely georegistration error is at different locations in the georegistered image. Press OK to georegister.
Manifold computes for a while and geregisters the image.

Because the reference drawing was in **Latitude / Longitude** projection the image now is also in this projection. We can see the image is in **Latitude / Longitude** projection because the latitude and longitude lines in the graticule are now straight horizontal and vertical lines. The internal scale of the image is slightly altered to preserve the original pixel aspect ratios as much as possible, so when seen by itself in an image window the image seems slightly compressed East to West.

We can open the error surface, called **Oz Error**, to see the root mean square error analysis. In the screen shot above we have used View - Display Options to specify a **Spectrum** palette to color the error surface so that different error values are more evident. Clearly, the central portion of the georegistration within the region covered by control points has low error in georegistration.

**Step 6: See the Georegistered Image in a Map**

Let's see how our georegistered image appears in a map together with a drawing that we know is exact. We import the **World** drawing published together with Manifold System for use as a base map. This drawing shows the countries of the world as areas. We then create a map using this drawing and open the map.
To show the Oz image in the map with the drawing we drag and drop the image into the map. It first appears as a layer above the drawing. We move the drawing layer above the image layer and then change the background color for areas in the drawing to transparent so that only the "edges" of the areas appear in the drawing. See the Transparent Areas topic for information on use of transparent colors as background or foreground colors with areas.

This allows the image to be seen through the areas in the drawing layer as seen above. We can therefore make a direct comparison between the shape of the georegistered image and the drawing that we know is very precise. Considering that we obtained the image by scanning a small image in a book the registration and accuracy are remarkably good. Very close inspection will show that the shorelines are slightly off in the extreme Southeast region of Australia and Tasmania, but the rest of the apparent shoreline is very well aligned to the drawing.

To see if the slight misalignment in the Southeast region is a result of the georegistration process we can drag and drop the Oz Error surface into the map as well. We change the opacity of the error surface to 50% so the lower layers can be seen through the error surface. See the Layer Opacity topic for information on changing the opacity of layers in maps.

We can see that a region of the lowest error values (in magenta color) passes right over the Bass Strait between Australia and Tasmania. It is most likely, therefore, that the lower correlation of the georegistered image and the known-accuracy drawing in the far Southeast is caused by lower accuracy in the original image. It was, after all, a small image printed in relatively low accuracy in a book to portray thematic information and was not intended for precision cartographic uses. The Winkels projection used in the book (a projection showing the entire Earth) also has its greatest distortion in the far Southeast.

**Step 7: Create Drawings by Tracing the Image**

We can use georegistered images for many purposes in maps just as they are. We can use them as backgrounds or as reference materials. For example, we can use the georegistered Oz image to see instantly in which climatic zones various cities might be located should we care to drag and drop a drawing of Australian cities into the map.

Another common use for georegistered images is the creation of new drawings by tracing. Images are intrinsically inefficient, so we can convert them into more efficient form by creating a new drawing and then drawing areas over the various regions shown by different colors of pixels in the image. See the Tracing topic for details.
To create a new drawing we begin by adding a new blank drawing to the map. Within the new drawing we can trace new areas. Let's create an area that covers the climatic region shown over central Australia. We turn off the error surface layer and then create a new area in the new drawing.

We create this layer using the Insert Area tool. After zooming into the map to a suitable degree we simply click along the edge of the dark yellow region seen in the image. The result is a new area (which we have formatted in yellow color with a speckled area style).

If we click off the image layer in the map we can see the world map layer together with the new drawing layer and the newly-created area. It's tedious, but if we wanted to we could use tracing to create new areas showing all the different climatic zones of Australia. We could add a field to the drawing used for tracing in which we could place a value for the different types of climatic zones. We could then use thematic formatting using the code for climatic zone to automatically color the new areas into different colors like those seen in the original image.

It's true that because the original Australia image is low accuracy we would have to interpolate by eye to draw the new climatic areas in the Southeastern part of Australia. However, the data set is obviously approximate and intended for a broad overview so such interpolation would be reasonable.

Notes

Many printed maps have graticules printed on them that can be used to repeat exactly the procedure above. The example map was a low quality, low accuracy, small map printed in a book. It was part of a map showing the entire world in Winkels projection, so the Australia portion of the map was quite distorted in projection. Nonetheless, the result of scanning and georegistration within Manifold was eminently useful. Had we started with a higher quality printed map, such as a USGS printed topographic "quad" map, our results would have been better.

Manifold and modern, high precision vector drawings used with Manifold are often much more accurate than printed maps. It is surprising to learn that a trusted atlas or even a chart intended for marine navigation may contain substantial inaccuracies. Such inaccuracies may arise from inaccurate data used to draw the map, distortions induced by the printing process or may even arise from poor quality projections. It is only recently that modern computers have made precision projection computations both automatic and routine.
When tracing areas such as the one created above there is a basic difference between the freehand, casual tracing possible in the region within Australia and the highly precise edge desired at the shoreline, where the newly-created area should precisely coincide with the detailed vector shoreline of the world map. The way to achieve a precise shoreline for the new area is to create the new area so that it substantially overshoots the shoreline. Next, since Australia is an area in the world map we can use the Australian area with the Clip with (Intersect) operator in the Transform toolbar to exactly trim the new area to the desired shoreline.

When working with scanned images of paper maps that show smaller regions of the world the graticule lines are usually aligned to "round" values using degrees, minutes and seconds notation. Part of the benefit of entering control points using a table is that we can enter values as read off the paper map using degrees, minutes and seconds notation and the table will automatically convert that for us to decimal degrees. See the Create a Table and Add Records topic for a detailed example showing how data entered in degrees, minutes and seconds can be automatically converted into decimal degrees.
Intersection Overlays

This is a small, but sophisticated, example that uses Transfer Rules, Split with, Clip with (Intersect) and Spatial Overlay commands. This example is an intermediate example that assumes basic familiarity with Manifold System.

Suppose we have a map with two drawings A and B that contain area objects that intersect each other. B contains green rectangles while A has blue triangles seen in 50% transparency so their overlaps with the B areas may easily be seen.

Both A and B have field names as seen in their tables above. The A drawing contains triangles and one field called Triangle with numeric values 1, 2, 3 or 4. The B drawing contains rectangles and one field called Square with text values Tom, Dick, Harry and Jane.

Our task is to create a new drawing that contains the spatial intersection of A and B, where all such objects have field values from both A and B. We also want the objects to be split up so that anywhere there is a boundary either from A or from B there will also be a boundary in the resultant drawing. This is a more complex task than simply finding the intersection of A with B where, say, just those parts of the B areas that are within the A areas are retained.

**Step 1: Get ready**

Like many GIS tasks, our task begins with a few preliminary steps to get organized. Begin by making a copy of B called B 2. Since we will operate on the B drawing we'll make a copy of the original drawing should we ever need it.
Right click onto the Triangle and the Square columns in the table windows and set the Transfer Rules for each to Copy in both directions.

We now need to arrange both drawings so that both A and B have fields of the same type. That is, we would like to equip the A drawing with a text field called Square and we would like the B drawing to have a numeric field called Triangle. With only two fields per drawing, we could quickly add one extra field to each drawing. However, if there are many fields involved we would like a faster way. Here is a fast trick:

Click on the A layer, click on one of the triangles and then Copy it. Click on the B layer and Paste the triangle into the B layer. This will add the fields from the A layer into the B layer. When pasted, the new triangle in the B layer will be selected. Since we don't need any triangles in the B layer, Delete it right away. The triangle will be gone but the new field structure will remain. Next, Copy a rectangle from the B layer and Paste it into the A layer. Delete the pasted triangle.
The result will be that both the A and the B drawings will now have the same fields. The newly-added fields will have default values (empty for strings, 0 for numbers) for the pre-existing objects. Note that the transfer rules established for these fields will be copied as well (which is one reason we set up the transfer rules as desired at the very beginning of this example).

**Step 2: Make boundary lines in A**

Since we will want to chop up the resultant areas by boundaries in both A and B, we’ll need some boundary lines in A with which to cut objects.

| [All Objects in A] | Boundaries | Apply |

Set up the transform toolbar as above and press **Apply**.

![Image of map with boundary lines drawn](image)

This creates lines in A that are the boundaries.

**Step 3: Cut areas in B**

We can now use the selected lines in A to cut the areas in B like wire through soft cheese.

| [All Objects in B] | Split with | [Selection in A] | Apply |

Make B the active layer by clicking on its layer tab. Load the transform toolbar as seen above and press Apply.

![Image of map with areas split](image)

This splits the areas in B along the boundary lines in A.
If we look at the table for B, we can see that new objects have been created. Because we previously set the 1:N Transfer Rules to Copy, each new area created from one of the four original rectangles inherited a copy of the Square value from its parent.

**Step 4: Create intersection of A and B areas**

We can now create the intersection of the A and B areas using the Clip with (Intersect) operator.

```
[All Objects in B]  Clip with (Intersect)  [All Objects in A]  Apply
```

Set up the transform toolbar as above and press Apply. This will use the areas in A (the lines are ignored) to clip the B areas down to only their region of intersection with the A areas.

As can be seen, the result is that all of the B areas that lie outside A areas are deleted. Because we previously split the B areas using the boundary lines in A, the edges of the A and the B areas are perfectly aligned.
There are fewer records in the table since some of the B objects that lay outside the A areas have been deleted.

**Step 5: Transfer A values to B areas**

The final step is to use Spatial Overlay to transfer the values of A areas to those B areas they contain.

We want to transfer all of the values of the Triangle field from the A drawing to the B drawing. However, we don't want to transfer the empty value of the Square field.

So, we right click on the Square field in the A drawing's table and uncheck the Transfer Column box. Press OK.
We then click on the B layer and choose the **Drawing - Spatial Overlay** command. Our **Source** is all objects in A and the **Target** is all objects in B with the method being to transfer values from areas to contained areas. This will transfer values from areas in A to all B areas that are contained by them. Press **OK**.

Like magic, the **Triangle** field in the B drawing will be populated with the appropriate value from whatever A area contains it.

If we turn off the A layer and then select some of the areas in B we can see that the result of our operation is indeed a group of areas that are the intersection of the areas in A and B that are correctly split along all boundaries that occurred either in A or in B.
From the table we can see that each area has correctly inherited the value of whatever original B field it had as well as the value of the A area that overlaid it.

**Notes**

Why did we take the time to create the same field structure in both drawings? The **Spatial Overlay** command can transfer data between layers in maps only when those layers have exactly the same field names and types, including the same transfer rules.

Why did we create boundary lines in A and then use the **Split with** command? Why not simply use **Clip with (Intersect)** directly? We could do that, if desired. The difference would be that the areas in B would not be split along the boundaries of the areas in A. For example, as executed above the procedure results in two areas in B in the locations marked by red arrows in the illustration below.

These were both part of the same rectangle initially. Had we omitted the **Split with** command these would still be part of the same rectangle and after the **Clip with (Intersect)** command would be one area.

**See Also**

- **Transfer Rules**
- **Transform Toolbar**
- **Boundaries**
- **Spatial Overlay**
- **Clip with (Intersect)**
- **Split with**
Transfer Contour Line Height to Points

This brief example uses Spatial Overlay to transfer the **Height** value of a contour line to points that are on that line.

![Contour Drawing](image)

Our drawing contains a line. A series of points are superimposed on the line. Perhaps these points were created from the line using the Transform toolbar Points transform operator.

![Contours Table](image)

Examining the drawing's table we can see that the line has a **Height** value of 810 while the points all have a **Height** value of 0. We use the **Type (I)** intrinsic field in the table to show which records are points and which record is the line. Our task is to transfer the height value of the line to all of the points that are on the line.

**Step 1: Specify Transfer Rules**

In Manifold the Transfer Rules for each column specify how the column's values are transferred between objects. In the table, right click onto the **Height** column head and choose **Transfer Rules**.

![Column Transfer Rules](image)

In the **Column Transfer Rules** dialog, choose **Copy** as the method for **1 to N** transfers. Transferring values from a line to many points that lie on it is a one to many or "1 to N" transfer. Press **OK**.

**Step 2: Make saved selections**

The most convenient way of using **Spatial Overlay** is to save the objects involved into saved selections.
Select the line and save it as a saved selection called line in the Selections pane.

This is a very simple example with only one line. If we had many lines (as in a contour map) we would select all of them and save them as a saved selection.

Select the points.

This is most rapidly done in a large map by setting select objects to Select Points and then pressing CTRL-T or choosing Edit - Select By Type.

Save the selection as a saved selection named points in the Selections pane.

**Step 3: Apply Spatial Overlay command**

From the drawing, choose Drawing - Spatial Overlay.
In the **Spatial Overlay** dialog, choose line as the **Source** and points as the **Target**. All saved selections will be available in the **Source** and **Target** list boxes. Choose **Line to contained points** as the **Method**. Press **OK**.


Like magic, the **Height** value from the line is transferred to all points on the line (that is "contained" by the line). In this example we had only one line with a value of **810**, so all points receive that same value. If we had many lines each with different values then each point on a line would receive that line's **Height** value.
Color Areas by Counts

This example shows how to color areas by the number of points they contain.

Suppose we have a drawing that contains areas and points. The areas show states. Each point represents a customer. Such a drawing may have been created as a result of geocoding customer addresses.

We would like to color the states as seen above so that states with more customers have a different color than those with fewer customers.

This is a simple use of transfer rules and spatial overlay. Please begin by reading the Transfer Rules and the Spatial Overlay topics.

**Step 1: Prepare Saved Selections**

We first begin by selecting all points and then creating a saved selection called Points. We then select all the areas and create a saved selection called Areas. See the Selection topic for information on making selections and using the selections pane to create saved selections.

Now we are ready to set up some transfer rules and then do a spatial overlay.

**Step 2: Specify Transfer Rules**
If we open the drawing’s table we see that the six area objects have the names of each state. We first add a new field called **Total**. To do this we right click on the **Name** column head and then choose **Add - Column**.

In the **Add Column** dialog we add an integer column called **Total**. Using a 32 bit integer column allows us to count large numbers of items (over 64,000). If we wanted to use even larger numbers we could have used a floating point **Type**.

This adds a new **Total** column, with default values of zero for each object.

We shall now make ready the **transfer rule** desired. We begin by right clicking on the **Total** column and choosing **Transfer rules**.
In the Column Transfer Rules dialog we set the N to 1 rule to Count. The 1 to N rule doesn’t matter, since we will be combining many points into the single area that contains them so all we care about is the “many to one” N to 1 rule. Using Count tells Manifold to count up the number of items that occur in each area.

Since we don’t want to accidentally eliminate the names of each state via transfer rules, we also right click on the Name field, choose Transfer rules and then uncheck the Transfer column checkbox. This will prevent the Name field from being affected by any operation (such as a spatial overlay) that uses transfer rules.

Step 3: Perform a Spatial Overlay

We can now click back onto the drawing and then choose Drawing - Spatial Overlay. Spatial overlays are used to transfer information from some objects to other objects.
As the **Source** we will use the saved selection called **Points**. This is all the points in the drawing. As the **Target** we will use the saved selection called **Areas**. This is all the areas in the drawing. As the **Method** we will use **Points to containing areas**. Press **OK**.

Because the transfer rule for the **Total** column was previously set up to be **Count**, this spatial overlay will count all the points within each area and will place the resulting sum in the **Total** column for that area.

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>Wyoming</td>
<td>4</td>
</tr>
<tr>
<td>23</td>
<td>Nevada</td>
<td>6</td>
</tr>
<tr>
<td>24</td>
<td>Utah</td>
<td>5</td>
</tr>
<tr>
<td>28</td>
<td>Colorado</td>
<td>6</td>
</tr>
<tr>
<td>37</td>
<td>Arizona</td>
<td>9</td>
</tr>
<tr>
<td>39</td>
<td>New Mexico</td>
<td>19</td>
</tr>
<tr>
<td>52</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>53</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>56</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>57</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>58</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Like magic, we can see that each state area has acquired a nonzero value in the **Total** field, the value being the total number of points counted within that area.

**Step 4: Thematic Formatting**

We can now use the **Total** field to thematically format the drawing as desired.
Right click onto the Area Background Color well in the format toolbar and in the resulting Format dialog use values like those above. The Yellows palette has been lightened.

The result is a drawing where areas are colored in darker yellow if they contain more points.

**Dot Density Maps**

Areas may also be thematically formatted using Style to show regions with greater concentrations of a particular variable with a denser pattern. To do so, choose one of the area styles that show dots or dashes and then thematically format the Size of that style.
The thematic format above used the **Total** field to thematically format the **Size** from 10 (large size = sparse pattern) to 2 (small size = dense pattern).

The result, using white color for background so that the **Style** is the dominant visual feature, is to show regions with greater counts of customers with denser pattern.

**Formatting by Density**

In the above examples we've used a **Total** field that simply counts up the numbers of points in each area. Although that gives us a total number of points in an area, it does not give us the density of the points. For example, a very large area like Montana with 20 points would have the same **Total** value as a very small state like Delaware with 20 points and thus be colored the same, even though the customer density (customers per square mile or square kilometer) of Delaware would be much greater.

We can color areas by density using Active Columns to compute density and then using the Active Column to guide thematic formatting.

Before we create an Active Column we will turn on one of the intrinsic fields we would like to use, the **Area (I)** field.

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>Wyoming</td>
<td>4</td>
</tr>
<tr>
<td>23</td>
<td>Nevada</td>
<td>6</td>
</tr>
<tr>
<td>24</td>
<td>Utah</td>
<td>5</td>
</tr>
<tr>
<td>28</td>
<td>Colorado</td>
<td>6</td>
</tr>
<tr>
<td>37</td>
<td>Arizona</td>
<td>9</td>
</tr>
<tr>
<td>39</td>
<td>New Mexico</td>
<td>19</td>
</tr>
<tr>
<td>52</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The table by default shows no intrinsic fields. To turn on an intrinsic field we choose **View - Columns**.
In the **Columns** dialog we check the **Area (I)** field and press **OK**.

That displays the **Area (I)** intrinsic field. To add an Active Column we right click on any of the column heads and choose **Add - Active Column**.

In the **Add - Active Column** dialog we choose a name for the new column, **Density**, and make it a floating point column. Press **OK**.

The default script for the Active Column is just a template that does nothing but which reminds us of what the structure of an Active Column script using VBScript should be.
We edit the script to the text above, which divides the total by the area.

For performance reasons, we left the default computation setting for the Active Column of on user request so the column is not automatically updated until we ask Manifold to recompute the Active Column.

We can do so by right clicking on the column and choosing Recompute.

The Density column will then automatically be recomputed.
We can now use the **Density** column in a thematic format to color the drawing by the unique values in the **Density** field.

That provides a nice illustration. However, we might want to color the drawing using **Natural Breaks** instead of unique values. To do so, we might want to adjust the Active Column script so that it results in larger numbers instead of the very small "E-11" density numbers currently used.

To do so, we rewrite the Active Column script slightly to multiply the densities by **1E+12** (one times ten to the twelfth) to make the density numbers bigger.
After using **Recompute** one more time we can see the numbers are in a more intuitively understandable scale.

We can now thematically format using the **Natural Breaks** method and the **Density** field.

The result is a very clear drawing that shows how the states involved fall into three levels of densities.

**See Also**

Selection
Transfer Rules
Spatial Overlay
Thematic Formatting
Combine a Surface and a Drawing in a Map

This example shows use of a surface layer in a map together with a drawing layer. We color the surface by height.

**Step 1: Locate and download the example files.**

The surface used is the **Montara Mountain** (California) 1:24K-scale DEM (Digital Elevation Module) downloaded from the USGS web site. This DEM is located just south of San Francisco and is provided by USGS in SDTS format.

The drawing used is the corresponding "DLG" (Digital Line Graph) USGS vector map, also in 1:24K-scale and also distributed by USGS in SDTS format. This file is also known as **Montara Mountain**, after the name of the paper "quad" map both the DEM and DLG cover. The STDS DLG quad contains several different data sets, including those for transportation, hydrology, boundaries and so forth. We have downloaded the files for transportation.

To locate the files, visit the USGS US Geodata web page, currently at [http://edc.usgs.gov/doc/edchome/ndcdb/ndcdb.html](http://edc.usgs.gov/doc/edchome/ndcdb/ndcdb.html) and drill down into the 1:24K DEM and DLG pages to find and download these maps. See the Where to Get Maps topic for notes on USGS sites. **Note:** USGS has recently embarked in a series of deals to give public data to private firms for their benefit. USGS DEMS are still available for free download, but you may have to download them from private firms who will also offer downloads for a fee in addition to the free USGS SDTS DEM downloads.

The maps will download as compressed files using **.tar.gz** archiving/compression. See the Import a Shapefile topic for notes on decompressing such files using WinZip (don’t forget to uncheck the “TAR file smart CR/LF conversion” box in WinZip’s Options). The result of decompressing an SDTS **.tar.gz** archive will be a folder full of various SDTS files. It is therefore wise to place each **.tar.gz** file in its own folder first and then decompress it.

In this example, within the **tmp** folder on our computer we have created two directories, a **montara_mountain_CA SDTS DEM** folder and a **montara_mountain_CA SDTS DLG** folder. Within the DLG folder we have further created a **transportation** folder. We then uncompressed the two **.tar.gz** files into these folders. SDTS data sets consist of so many files that we will be glad we took the time to keep them well organized through sensible use of folders in our Windows file system.

**Step 2: Import into Manifold**

Launch Manifold. We will begin by importing the surface. Choose **File - Import - Surface** and choose SDTS Files in the **Files of type** box. **Note:** SDTS files can contain surfaces and drawings so Manifold's SDTS importer appears both under **File - Import - Surface** and **File - Import - Drawing**. Regardless which menu path is chosen the SDTS importer will correctly detect the presence of surfaces or drawings in the file and import the data correctly as either a surface or a drawing.

Browse over to the folder in which the downloaded SDTS **.tar.gz** file was extracted. Although Windows Explorer would show us many SDTS files, only one of them is a **.ddf** file that summarizes the content of the data set. Only this **.ddf** file is shown in the import dialog. We choose the **.ddf** file in the folder and press **OK**.
In the **Import SDTS Data Set** dialog we leave the default choices as they are and press **OK**.

The result is that new surface, terrain and comments components appear in our project. We will next import the drawing.

![Import SDTS Data Set dialog](image)

Once again we choose **File - Import - Drawing using SDTS Files**. We browse over to the folder where we extracted the SDTS DLG drawing files. For this example, we will use the roads files from the transportation layer. We choose the `.ddf` file in the transportation folder and press **OK**.
We will use the default settings once more and press \textbf{OK}. Later in this example we will see that using the default settings is a minor mistake. For now, we will make the mistake and then later correct it.

The result in the project pane is the appearance of more components, including a drawing component. Note how imports from SDTS are self-documenting.

\textbf{Step 3: Create a map}

We choose \textbf{File - Create - Map} or use the \textbf{Create} button in the project pane to create a map.
In the **Create Map** dialog we check the surface as the component to be used to create the map. We could have used both, but when using new data sets we like to create maps using one component and then drag and drop the other components into the map. That assures the map will be created using the projection of the one component used to create it. This is important when working with potentially large surfaces since map display will go fastest if the map uses the same projection as any large component it contains.

If we double click open the map thus created we see it contains one layer, the surface used to create the map. The surface is seen using default display options.

**Step 4: Color the surface by height**
To provide additional visual cues for reckoning elevation as well as to provide a more striking image we will color the surface by the height of each pixel.

To do this, choose View - Display Options to launch the Display Options dialog. In the dialog choose the Spectrum preset for the Palette and press Apply. Leave all other options as indicated in their default settings. The settings ask Manifold to show the scene with shading as if the Sun were in the East at a 45-degree angle above the horizon. This is not exactly a realistic setting since the Sun is only directly in the East at Sunrise when it is well under 45 degrees above the horizon. However, the settings provide a pleasing visual effect. Press OK.
Well... that's certainly a striking image! Manifold has colored the surface using the palette intervals and colors indicated.

**Step 5: Drag and drop drawing into the map**

We will now add the drawing to the map.

Drag and drop the drawing from the project pane into the opened map window. It will appear as a layer just above the surface.
Well... that's certainly an ugly effect! What happened? The drawing as currently structured contains numerous areas that completely cover everything below so the surface cannot show through. In addition, the drawing contains numerous points that are shown by default as small circles. To see just the roads we ideally would want only lines in this drawing.

We also have points and areas because of the small mistake we made when importing the drawing.
If we recall the SDTS import dialog we used to import this drawing, we simply used default settings for import. Default settings check all the Modules boxes, which happen to include modules for points, lines and areas. If we only wanted lines we should have unchecked the points and areas boxes (scrolled out of sight in the illustration above) and imported only the lines modules.

**Step 6: Remove areas and points from the drawing**

To remedy the problem we will select all points and areas in the drawing and delete them.

To do so we push in **Select Areas** and **Select Points** in the Select Objects buttons of the selection toolbar and we push out **Select Lines**.

We then choose **Edit - Select By Type** to select all objects in the active drawing that are chose by the Select Objects filter buttons.
All areas and points in the drawing are selected. To delete them we press the **Delete** key on the keyboard or choose **Edit - Delete** in the main menu.
When the areas and points are deleted only the lines appear in the drawing layer. This allows the surface to be seen.

**Step 7: Alter surface transparency**

The preceding image is certainly spectacular. However, the intensity of the colors in the surface make it difficult to see the road lines. If we wish to provide greater emphasis to the road lines we can de-emphasize the surface by making it partially transparent. To do so, we open the Layers pane using `SHIFT-ALT-L` or `View - Panes - Layers` from the main menu.

Press the **Opacity** button in the layers pane to show opacity for each layer. Double click into the 100% opacity value for the surface layer and change it to 50%.

This immediately de-emphasizes the surface by allowing the white background to shine through it. The result is a display that shows roads to a much greater degree than before.

We can also combine different levels of surface opacity with various formatting options for the drawing layer.
Here we have zoomed into the western edge of the map. The opacity of the surface has been set to 75%. The lines in the drawing layer have been set to size 2 to make them thicker. Amazing detail, isn't it? One can see how the good citizens of this part of California like to place their homes on hillsides. Note how a road in the center of the image crosses what appears to be a stream channel. If we like, we can import the hydrography layer from this SDTS file and see if it shows a stream in that location.
Here we see a view zoomed into the eastern edge of the map, where the road crosses over the valley of the San Andreas Fault. The flat area is the San Andreas reservoir that provides water storage for the city of San Francisco. The reservoir running from upper left to lower right and the valley down to the lower right mark the location of the fault itself. The San Andreas Fault was the locus of the earthquake that destroyed San Francisco in the early part of the 20th century.

**Step 8: Add a second drawing to provide context**

Readers familiar with the Bay area might wonder where exactly is the Montara Mountain quad we see. To provide context, we can add another drawing to the map.

We will import the `bay_land_eg.mif` drawing used in the My First Map example and drag and drop it into the map. It appears above the other layers so we will right click on its layer tab and use Order to send it to the bottom of the layer stack.

We can now see the surface and roads drawing in position above the `bay_land_eg` drawing.
If we zoom into the lower left corner of the surface we can see it is located at Half Moon Bay. For this illustration and the one above lines in the roads layer have been set to a size of 1. Opacity of the surface is still at 75% so the gray areas of the bay_land_eg drawing can partially show through.

For Advanced Users

In this example we’ve combined drawings and a surface in an elementary way within a map. Note that part of the surface extends beyond land areas because USGS provides DEMs as rectangular data sets. Suppose we wanted to remove that part of the surface that did not extend over land?

One way to do this is to copy the surface and paste it as an image and to select those pixels that are over water and delete them (turning them into invisible pixels). The trick is to select only those pixels. To do so we can use a drawing as a guide. We do this by converting the drawing into an image using the Tools - Make Image command. We then force the resulting image into only black and white and (after matching it to the target image derived from the surface with Match ) use it as a selection mask to select only those pixels over water. We can then delete them.

See Also

See the Overlays topic for use of these same imported drawings to overlay the 3D terrain with lines from the roads drawing.
Adding a Legend

This topic shows how to add a legend to a surface. It shows simple methods for customizing the appearance of legends. The procedures shown here apply to creating and customizing legends for other components, such as maps and drawings.

**Step 1: Import a surface and color the surface**

Use the procedure set forth in Combine a Surface and a Drawing in a Map to import the **Montara Mountain 1:24K-scale USGS SDTS digital elevation module**. Open the surface and use the View - Display Options dialog to color it with the **spectrum** palette.

![Display Options dialog](image)

Click into each of the intervals in turn to edit the values for each color to an even number, such as 0, 300, 600, 900, 1200, 1500 and 1800.
The result is a surface that has been colored by the thematic format (palette) given in the display options dialog.

**Step 2: Show the legend**

Launch the **Legend** dialog with **View - Legend**.

Check the **Show Legend** box and choose **None** in the **Align** box. This will turn on the default legend. Because there is no specified alignment we will be able to drag the legend to any position in the surface window. If we press **OK** at this point the legend will appear in the surface window. However, before pressing **OK** we first will make some minor adjustments.
The name of the surface (and thus, the first line in the legend) is still the same long name automatically given to the component on import. We will improve this. Check the Customize legend box and then click on the first line in the legend pane to select this element for editing.

Change the string in the Text box from [Component] to Elevation. Note that the first line in the legend preview pane also changes. This changes the caption from an automatically generated escape sequence that prints the name of the component to the specific text string Elevation.
Push OK and the legend appears in the surface window. We can drag it to any position in the window desired.

**Step 3: Modify the legend color samples**

The legend created by default is fine for many purposes. If desired, we can change the style of the legend by changing settings in the **Legend** dialog.

We begin by clicking on the main element to highlight it, a dynamic element that gives formatting for the surface.

With the element highlighted we press the **Flatten** button to convert the dynamic element into a series of individual elements.
We can now individually manipulate the legend elements. We begin by highlighting the **Height** text element and then deleting it.

We will now change the shape of the samples used in the legend entries. We begin by clicking on the **1800** element to highlight it. In the **Type** box choose **Color: Square**.
This changes the style of color sample used from a wide rectangular box (the default) to a smaller square box.

We can click on each element in turn and set the **Type** for each to **Color: Square** to change all of the samples to this type.
Press **OK** and the legend appears using the new types.

**Step 4: Show two columns in the legend**

So far the legend shows all entries in a single vertical column. We can use more than one column in a legend for a more horizontal legend style.

Open the legend once more with **View - Legend**. Click on the **900** element in the legend preview pane and click the **Add Element** toolbar button.
This adds a new element just above the previously highlighted element. By default, the new element is a Text line.

To change the new element, we change the value in the Type box from Text to Vertical Separator.
The text element shown in the legend preview pane changes to a **vertical separator** element.

If we now press **OK** we see that the legend is drawn using two columns with the column break happening where the vertical separator element was inserted.

If we like, we can use a vertical separator line to set off columns. To do so, we open the legend again with **View - Legend**.
Click on the **column** element in the legend preview pane and choose **Vertical Separator with Line** as the **Type**. The element changes to **vertical separator with line**.

Press **OK**. The result is a legend in two columns where a vertical line separates the two columns.

**Step 5: Changing margins in a legend**

We might want to vary the amount of margin space between the contents of the legend and the legend border. This value is set by the **Margin** box. The default value is 8 points.

To change the margin, launch **View - Legend** and change the **Margin** value to 15 points. Press **OK**.
The legend reappears with greater margin space between the contents and the outside border. Additional space also separates the columns from the vertical separator line.

**Step 6: Create a horizontal legend style**

We can create a legend where every element is in a separate column for a purely horizontal style.

Open the legend with **View - Legend**. Set the **Margin** back to the default value of 8 points.

![Legend settings](image)

Click on the entries in the legend preview pane and use the **Move Up** and **Move Down** keys to rearrange them so the topmost is 0 and the lowest is 1800.

Next, add a **Vertical Separator** between each element. This places each element in its own column.
To tighten the legend arrangement, change the Align text setting to **Left**. This will bring the labels closer to the color sample boxes. Press **OK**.

The result is a pleasant legend that is horizontally oriented. Such legends take a few extra mouse clicks to create but are well suited for use within print layouts that use paper in landscape orientation.

**Step 7: Add a legend to a print layout**

We will frequently want to add legends to print layouts. The following layouts use small windows to better fit into this documentation. In real life we would use larger windows for more convenient operation.

Before creating the print layout we hide the legend currently displayed for the surface by choosing **View - Legend** and in the **Legend** dialogue un-checking the **Show Legend** box. If we create a print layout from a component where the component is already showing a legend the display in the print layout will automatically show that legend as well within the component.

For the sake of this example, we prefer to create an independent print layout element showing a legend so that we can move it about in the layout independently of any layout element that shows the surface.
Create a print layout using the Montara Mountain surface. We open the layout, and **CTRL-ALT-click** onto the surface to select it for editing. We drag the lower edit handle up to resize the element so that it does not fill the entire page.
This leaves some white space in the bottom of the page where we can place a legend.

Choose the Insert Legend tool.
Draw a box with the tool where the legend is to appear, dragging open enough space for a legend to fit.
The legend appears in the print layout. Manifold knows which legend is associated with the only component that appears in the layout. If multiple components appeared in the layout, then when we used the **Insert Legend** tool the system would have raised a secondary dialog allowing us to pick the parent component for the legend.
If desired, we can **CTRL-ALT** click on the legend and move it to any position in the layout independently of the surface component in the layout. Click anywhere outside the legend to deselect it.
If we zoom in we can see that the legend is the same as created in the previous steps of this example.

**Changing the Legend in the Layout**

Note that legends are configured based upon their parent component. If we would like to change the appearance of this legend, we can do so using either of two methods:

- We can right click on the legend in the layout, choose **Edit** and then configure the legend as desired. Double clicking the legend will also open it for editing.
- We can open the surface in its own window and then use **View - Legend** to configure the legend as desired.

Any changes made in the layout to the legend will appear in the legend in the surface window and any updates made in the surface window will appear in the layout window legend as well.

**See Also**

- **Adding Legends**
- **Legends**
- **View - Legend**
- **Legends, Scale Bars and North Arrows in Layouts**
Graticule Labels in a Print Layout

In print layouts we frequently would like to add latitude and longitude labels to the edges of a map or other component in the layout.

Creating graticule labels is a two-step process in Manifold: first a graticule is configured for the element within its own window and then the graticule is used to guide the creation of labels in the layout window.

In this example we create a layout using one element, a drawing of Mexico, with graticule labels on its border.

Step 1: Configure a graticule in the drawing

We begin by importing a drawing of Mexican provinces, using either the Mex boundaries.mfd or mexico_eg.mif sample files on the Manifold CD.
Examples

To show a graticule we open the drawing and choose View - Graticule command.

In the Graticule dialog we check the Show graticule box and then use the Suggest button to get a suggested range of latitudes and longitudes to use. We then round these off by eye to go from longitude -120 to longitude -85 and from latitude 10 to latitude 35. We use a spacing of 5 degrees for both latitude and longitude. Press OK.

The result is a graticule that neatly covers Mexico with a reasonable grid of latitude and longitude lines. How did we get the numbers we entered into the Graticule dialog? Through experience we know that a 5 degree grid is about right, with graticule lines neither too far apart nor too close together. Too get the other values we started with the Suggest button and then rounded them to the nearest 5 degree value. By trying a few different values (it only takes a few seconds to see how the graticule looks in the drawing) we quickly zeroed in on what values worked well to cover the Mexico drawing.

**Step 2: Create a layout and turn on graticule labels**

Next we create a print layout by right clicking into the project pane and choosing Create - Layout. We create the layout using the Mexico Drawing component.
Since we left the graticule turned on in the drawing, the drawing appears in the layout with a graticule overlaid. To add graticule labels we right click onto the drawing in the layout and choose Properties.

In the Properties dialog we choose coordinates (graticule) in the Border option. We also check the Deg-min-sec format box to use a degrees, minutes and seconds format for the latitudes and longitude labels. Press OK.
Examples

Labels now appear on the border of the drawing.

If we zoom into the layout we can see that the drawing now has a rectangular borderline drawn about it. Where the edges of the border are intersected by the graticule lines a tick mark appears with a label giving the longitude or latitude for that graticule line. Note that the labels are written using degrees, minutes and seconds notation, which appends a N, S, E or W to indicate North and South latitudes or East and West longitudes. Because the graticule is aligned on round numbers every five degrees there is no need for minutes or seconds in the labels, so we've used the default Rounding of 0.

Variations

Suppose we would like graticule labels on the edges of the drawing but we don't want to show the graticule itself? That's easy to do. Right click on the drawing element in the layout and choose Properties.
In the Graticule option change the default auto setting to hide. The auto setting will show the graticule if it is shown in the drawing and will hide it if it is hidden in the drawing. The hide setting will always hide the graticule. Press OK.

The result is that the graticule no longer appears in the layout, but the graticule labels still appear. This is often a useful effect in maps of small regions where latitude and longitude lines form a very rectangular grid. However, it can be confusing in maps where graticule lines are highly curved because it is not clear to the person using the map how to extrapolate from the labels on the edges into the interior of the map.

To get an example of how graticule labels look in the default decimal degrees format we can right click on the drawing element in the layout, choose Properties and uncheck the deg-min-sec format box.

In a zoomed-in view we can see that the labels along the upper edge have dropped the appended W and now are negative numbers to indicate West longitude.

See Also

Layouts
View - Graticule
Align Items in Layouts

This example presents a quick sequence of alignment commands that are used to align text in a print layout using tools on the Alignment toolbar. The print layout contains a map of Europe and four text boxes that are arranged in the four corners of the layout. The task is to move the texts so they neatly line up with each other.

Before we begin aligning elements, let's change the selection style so that selected items are shown with a red border only and not with the default, filled-in red dot pattern used for selections.

Click open the Selection Style menu.

Choose Border.

Step 1: Select and align leftmost text

We'll begin by aligning the lower left text so that it is even with the upper left text.
Click on the lower left text to select it.

CTRL-ALT click on the upper left text to select it for editing. Selecting another item for editing using CTRL-ALT click will not deselect the lower left text. When an item is selected for editing, it becomes the primary selected item.

Press the Align Left button on the Alignment toolbar to move the lower text box to the left to align to the primary selected text box. The text will move to the left.
Step 2: Select and align upper text.

Next we will move the upper right text up to align with the "Europe" text in the upper left. We begin by **SHIFT** clicking on the upper right text. This will select it without deselecting any other items.

We then **CTRL** click on the lower left text to deselect it. A **CTRL** click toggles back and forth between selected and unselected without affecting other items.
We now have only two items selected, with the upper left text being the primary selected item.

Press the **Align Top** alignment button to move the upper right text upward so it is even with the primary selected item, the "Europe" text in the upper left.

**Step 3: Select and align rightmost text.**

We will now align the two rightmost texts using a similar procedure.
CTRL-ALT click on the lower right text to make it the primary selected object.

CTRL-click on the upper right text to deselect it. Both texts on the right are now selected with the lower right text being selected as the primary selected object. We will move the upper text to the right to align with the lower text.
Push the Align Right button to move the upper text box to the right to align with the primary selected text in the lower right.

**Step 4: Select and align lower text.**

The final step in this example sequence will be to align the lower texts. We will move the rightmost text down in alignment with the lower left text.
We **CTRL-ALT** click the lower left text to select it as the primary selected object and then we **CTRL** click on the upper right text to deselect it. We now have the lower left text as the primary selected item and only one other item selected, the text box that is to be moved down.

Press **Align Bottom** to move the text on the right down so that it is aligned to the lower edge of the text box at lower left.

The result is a symmetric layout with evenly aligned text.
Text Box Formatting and Alignment

The procedure above neglected to mention an important point in order to show a simple example of how to use
the alignment tools. Because text boxes can be drawn much larger than the text they contain it is important to
align the text within the text box using the Format Toolbar controls in a way that anticipates how the alignment
commands will be used.

The alignment tools in the Alignment toolbar will align the edges of the text boxes. If we want two texts to be left-
aligned to a common vertical margin, we should not only align the text boxes but we should also use the Format
toolbar to left align the text within those boxes. This can be done either before or after the above procedure.

For example, if we look at the label in the lower left corner, the default format setting is that the text is centered
within the text box. This leaves some space between the left edge of the text box and the beginning of the actual
text. If we used the Align Left command with this text box to align it to the left edge of another text box, the text
boxes will be aligned but the text within this text box will appear slightly indented due to the space between the
left edge of the box and the beginning of the actual text.

We can remedy that by CTRL-ALT clicking on the text box to select it for editing and then using the Format
toolbar to left align the text within the box.

In the Format Toolbar we can click Align Left to left align the text within the text box.

We can also click Align Bottom to align the text to the bottom of the text box, a good idea when aligning single
lines of text so that they all share the same base line for the characters.

The left alignment is visible, but the bottom alignment is not because the text box is very small compared to the
size of the text within it.
Likewise, the texts that are to be aligned on the right side of the layout should be right aligned within their text boxes. We can do this by \texttt{CTRL-ALT} clicking each one to select it for editing.

Each will have the default formatting of being centered within the text box.

We can click \texttt{Align Bottom} to bottom align them and then click \texttt{Align Right} to right align them within the text box.

If the labels on the right are aligned to the right within their text boxes, when the text boxes are aligned to a common margin on the right side the texts will also be aligned to that margin.

\textbf{See Also}

\begin{itemize}
  \item Alignment Toolbar
  \item Format Toolbar
  \item Layouts
\end{itemize}
Create a Print Layout from a Table

In this example we create a print layout from a table using the sample *Mexico_eg.mif* drawing from the Manifold CD. To simply print a table we could always just open the table and choose File - Print; however, using a layout can give us better control over the appearance and format of the printed table.

**Step 1: Import drawing and configure table**

Import the *Mexico_eg.mif* drawing that is found on the Manifold CD.

![Mexico_eg Drawing](image1)

Open the table.

![Mexico_eg Table](image2)

Note that the table includes the **ID** field that gives the object ID. If we wish to print the table to get a record of the attributes we may decide to not print the **ID** column. To get rid of the **ID** column we right click onto the **ID** column header and choose **Hide**.

![Mexico_eg Table](image3)

That's better!

**Step 2: Create a print layout from the table**

A fast way to create a layout from the table is to right click onto the table in the project pane and choose **Create - Layout**.

![Create Layout](image4)

Doing so automatically opens the **Create Layout** dialog with the table that was right clicked already highlighted to use as the component from which the layout will be created.
Press OK to create the layout using the **Mexico_eg Table** component. This creates the layout using the table as the main element in the layout.

A new layout will appear in the project pane. It is shown indented under the table that is the main element in the layout. Double click the layout to open it.

The layout opens using the default page setup for the default printer, with as many pages used as are needed to display the contents of the table. The table is centered vertically. Since tables are often printed using multipage layouts, it is a good idea to review the Multipage Layouts topic to understand how selection and editing is done when more than one page is used for a layout.

**Step 3: Adjust the table**

To change any settings we must select the table for editing by **CTRL-ALT** clicking it.
When the table is selected for editing it will be shown in selection style (we've changed the selection style to **Border** so that the table is more clearly visible) and edit handles will appear.

The red outline shows the extent of the layout element with which the table appears. To reduce the white space at the top and bottom of the table we can resize the layout element by dragging the bottom middle edit handle upwards.

Making the table element less tall will reduce the white space at the top and bottom of the table. Note that the table text will occupy the same amount of space regardless of how we resize the table element box because the size of the table is fixed by the physical size of the font used for the table text. The font size may be changed using the Format toolbar, which is enabled when the table is selected for editing.

**Step 4: Adjust page orientation and number of pages used**

From the illustration above it is clear that using a **Landscape** orientation for the page might possibly allow the entire table to be printed on a single page.
If we use File - Page Setup to change the paper orientation to **Landscape** we can see that this does, in fact allow the entire table to fit on a single page. We can now adjust the layout to use only one page by using the **Layout - Pages** dialog.

Change the **Pages** numbers to 1 in both boxes for a single page.

The result is a table printed on a single page in **Landscape** orientation.
Why go through so many steps just to print a table? Using a layout allows us to add other text elements to the layout, for example, a title block with copyright or other information or perhaps other text boxes that automatically print page numbers, time and date, the names of files or components used or other information.

We can also adjust the appearance of the table by choosing the font and other items. See the Printing Tables topic for a full list of properties that may be adjusted.

Note that when adding or deleting columns Manifold will try to keep the table centered within the table element. Therefore, resizing the table element box may be necessary after adding or deleting columns.

See Also

Layouts
Multipage Layouts
Printing Tables
**Editing a Surface for Visual Effect**

Simple editing techniques can be used with surfaces to provide specialized visual effects. In this example we will highlight a band of elevations between 500 and 1000 feet in a surface. Illustrating a specific band of elevations in a surface is often desired for planning presentations. For example, we may wish to show regions in microclimates that are above valley fog but below mountain snows. We will do this by creating an image from the surface, colorizing the image and showing it overlaid upon the surface in a map.

**Step 1: Prepare components to be used**

Import the Montara Mountain surface as described in the Combine a Surface and a Drawing in a Map topic. Use View - Display Options to color it with the **Altitude** palette.

In the project pane, copy the surface and paste it to create a second surface component. Open this copy in a surface window.

**Step 2: Cut out a band from 500 to 1000 feet**

With the surface opened, in the Layers pane turn on the **Background**. Use View - Properties to set the background of the surface to a blue color.
Next, use the **Threshold Lower** transform operator to force all pixels at 500 feet or below to the 500 value.

![Tool Properties](image)

Click on the **Select Touch** tool and in the Tool Properties pane set the tolerance to zero.

![Select Touch](image)

**Shift**-click onto the region of 500 foot elevations to select all such regions in the surface.
We can now choose **Edit - Delete** to delete all pixels at this level, allowing the blue background color to be seen. As can be seen in the illustration above, this is a cool way to achieve "waterline" effects. If we had not turned on the **Background** we would not see the blue background color. [For fun, to see a surface "flood" try using **Threshold Lower** repeatedly with settings of 100, 200, 300 and so on to delete sequential heights of pixels.]

We now run **Threshold Upper** with a setting of 1000 to force all pixels above this level to 1000.
We can then Shift-click into the region of 1000 foot elevations using the Select Touch tool to select all such pixels, and then once more we can use Edit - Delete to delete all such pixels.

What's left is a band of surface pixels that are between 500 feet and 1000 feet.

**Step 3: Create an image**

Since more graphics arts effects can be applied to images than to surfaces we will now make an image from this surface. To do so, launch the Tools - Make Image command.

This saves the surface as the same sized image. When we open the image it will be visually identical to what we saw in the surface window.

**Step 4: Modify the image**

Open the image. It appears the same as the surface.
Using Touch selection, SHIFT-click onto the blue pixels to select all blue pixels in the image. Press Edit - Delete.

This deletes all of the blue pixels in the image.
We will now force all colored pixels to white by using the Image - Adjust - Hue / Saturation command and by turning **Lightness** all the way up.

**Step 4: Show the image in a map with the surface**

We are now ready to use the image in a map with the original Montara Mountain surface.

Create a map with the image in an upper layer and the surface in a lower layer. Right click on the image layer tab and set the opacity of the surface to 50%. It appears as a whitish layer. We would like to accentuate the color of the image to make the position of the band between 500 feet and 1000 feet more obvious.
To do this we click on the image layer in the map and choose the Image - Effects - Colorize command from the main menu. We move the Blue slider all the way to the left and the Red slider all the way to the right. The Green slider is almost all the way to the left. This colorizes the image layer into a red hue. If we do the Colorize operation with the Preview box checked we could interactively try different colors to get the desired effect.

As seen in the illustration above, we've tinted precisely the band of pixels between 500 and 1000 feet.

Notes

Why did we set the background color to blue? We did this so that the image created from the surface has a background of blue pixels. Had we not specified a blue background, the image would have been created with black pixels in the "transparent" regions of the surface. If we would use SHIFT-click to select all black pixels in the surface, this would have selected a few pixels in the deeper shadows within the band of remaining surface pixels. By using a background color that is not at all like any of the colors in the surface we make it easy to select the background color only and not accidentally select any pixels from the remaining surface.

Why did we use an image in a layer to color part of the surface? Could we not have simply altered the Altitude palette used to color the surface so that regions between 500 feet and 1000 feet were colored in red? We could have done that, but using a palette will normally blend colors in between altitude ranges. Using an image as we have done allows us a sharper control over appearance. It also demonstrates how images and surfaces can be combined in maps to achieve desired effects.
Copy and Paste: Image / Table / Drawing

In this example we will copy an image and paste it as a table. We will then copy the table and paste it as a drawing. This topic provides an example for the concepts discussed in the Copy and Paste As topic. If you've jumped here directly from that topic please note that this example uses concepts (such as RGB values) introduced in the Images topics.

**Step 1: Copy an Image**

![](head.png)

Let's begin with a project that contains an RGB image called head.

If we open the head image in a window, we can see it is a small image we have created by cropping our sample bronze image to show just the head of the bronze monument.

If we zoom into this image we can see that like all images it is made up of pixels. Each pixel appears at a given location and consist of color made up of Red, Green and Blue channel values.

To copy this image, we right click on the head component in the project pane and choose Copy from the context menu.

**Step 2: Paste As a Table**

We can then right click in the project pane and choose Paste As - Table from the context menu.
The Paste As Table dialog appears. Because Manifold knows we have copied an image and are now trying to paste it as a table, the dialog is loaded with choices of information that may be extracted from the image for each pixel and pasted into the table as fields. See the Intrinsic Fields in Tables topic for a description of available fields. We choose the Red (I), Green (I) and Blue (I) channel values as well as the Longitude (I) and Latitude (I) fields (scrolled out of sight in the illustration above).

Checking the Longitude (I) and Latitude (I) values indicates we want the position values of the pixels to be extracted as geographic coordinates.

The result of the Paste As - Table operation is a new table in our project called Head 2. The system tries to use a name that is similar to the copied component in a case-insensitive way. Since "Head" is already used in the project the system adds the number 2 to create the new default name.

<table>
<thead>
<tr>
<th>Red (I)</th>
<th>Green (I)</th>
<th>Blue (I)</th>
<th>Longitude (I)</th>
<th>Latitude (I)</th>
</tr>
</thead>
<tbody>
<tr>
<td>95</td>
<td>135</td>
<td>205</td>
<td>0</td>
<td>1.931777612...</td>
</tr>
<tr>
<td>91</td>
<td>139</td>
<td>205</td>
<td>8.98315284...</td>
<td>1.931777612...</td>
</tr>
<tr>
<td>92</td>
<td>130</td>
<td>201</td>
<td>1.79662056...</td>
<td>1.931777612...</td>
</tr>
<tr>
<td>87</td>
<td>137</td>
<td>206</td>
<td>2.64946585...</td>
<td>1.931777612...</td>
</tr>
<tr>
<td>92</td>
<td>137</td>
<td>205</td>
<td>3.99326113...</td>
<td>1.931777612...</td>
</tr>
<tr>
<td>92</td>
<td>135</td>
<td>203</td>
<td>4.49157642...</td>
<td>1.931777612...</td>
</tr>
<tr>
<td>89</td>
<td>137</td>
<td>203</td>
<td>5.3689170...</td>
<td>1.931777612...</td>
</tr>
<tr>
<td>88</td>
<td>137</td>
<td>204</td>
<td>6.28206999...</td>
<td>1.931777612...</td>
</tr>
<tr>
<td>88</td>
<td>140</td>
<td>206</td>
<td>7.1865227...</td>
<td>1.931777612...</td>
</tr>
<tr>
<td>88</td>
<td>138</td>
<td>206</td>
<td>8.4647764...</td>
<td>1.931777612...</td>
</tr>
</tbody>
</table>

If we double click open this table we can see it contains rows with Red (I), Green (I), Blue (I), Longitude (I) and Latitude (I) fields for each row. Each row in the table came from one pixel in the image and contains the various...
field values for that pixel. The **Longitude (I)** and **Latitude (I)** values are the exact longitude and latitude coordinates of these pixels.

**Step 3: Copy the Table**

We will now copy the table. To copy the table, we right click on the **Head 2** component in the project pane and choose **Copy** from the context menu. We can then right click on any empty location in the project pane and choose **Paste As - Drawing**.

**Step 4: Paste As a Drawing**

This launches the **Paste As Drawing** dialog. Manifold knows we have copied a table and are attempting to paste it as a drawing, so the dialog is loaded with choices that make sense for pasting this particular table. Since the table has five fields, those fields are available for pasting into the drawing's table. We will check only the **Red (I)**, **Green (I)**, and **Blue (I)** table fields to paste as fields into the drawing's table.

When pasting a table as a drawing, for coordinates we could use any numeric fields as coordinate fields. Normally the fields used are those that have longitude and latitude values in them. [The option to use any two numeric fields is occasionally used to create a drawing that is, in effect, a scatter diagram of tabular data using two non-geographic data fields as X and Y coordinates.]

By default, if there are numeric fields named "Latitude" and "Longitude" Manifold assumes we would like to use those fields for Latitude and Longitude coordinates and will pre-load the **X / Longitude** and **Y / Latitude** boxes with those field names. In this example our longitude and latitude values are saved in the **Longitude (I)** and **Latitude (I)** fields so we will use them for longitude and latitude.

The result of the **Paste As - Drawing** operation is that a new drawing and its associated table appear in the project pane. As before, to create the name of the new drawing Manifold simply appends a new number to the name of the pasted component if it otherwise duplicates a component name already in use. In this case, the name for the new drawing becomes **Head 3** and the name of its associated table is **Head 3 Table**.
Double clicking open this new drawing shows that pasting the table as a drawing has created a regular array of points. We have formatted these points as small squares and have colored them using thematic formatting so that the color of each point is based on the value of the **Blue (I)** field. Each point came from a row in the table. Each row in the table, of course, came from one pixel in the original image. Each point in this drawing therefore represents a point that was derived from a pixel location in the image.

<table>
<thead>
<tr>
<th>ID</th>
<th>Red (I)</th>
<th>Green (I)</th>
<th>Blue (I)</th>
</tr>
</thead>
<tbody>
<tr>
<td>69154</td>
<td>95</td>
<td>133</td>
<td>205</td>
</tr>
<tr>
<td>69155</td>
<td>93</td>
<td>139</td>
<td>205</td>
</tr>
<tr>
<td>69156</td>
<td>92</td>
<td>130</td>
<td>201</td>
</tr>
<tr>
<td>69157</td>
<td>87</td>
<td>137</td>
<td>206</td>
</tr>
<tr>
<td>69158</td>
<td>92</td>
<td>137</td>
<td>205</td>
</tr>
<tr>
<td>69159</td>
<td>92</td>
<td>135</td>
<td>203</td>
</tr>
</tbody>
</table>

If we click open the **Head 3 Table** table that is associated with this drawing we can see that each record in the table has four fields. The fields are an ID field that links it to a point in the drawing, plus fields the **Red (I)**, **Green (I)** and **Blue (I)** table fields that represent the Red, Green and Blue channel values for the pixels in the original image. Each row in this table is linked to a point in the drawing and represents color data derived (via two **Cut** and **Paste As** operations) from a pixel in the original image.

We can see this is the case by formatting the drawing to increase the size of the square symbols used for points until they merge together at the current zoom level. We can also use thematic formatting to color the points using a range of blue color from dark to very light that is controlled by the content of the **Blue (I)** field.

The result of the formatting applied is to show the drawing (at the current zoom level) as a pixelated rendering similar to the image. The effect disappears as soon as we zoom in or out since it depends upon having the size
of the point symbols fitting together. We use it here not as a replacement for use of images but rather to show how the information content in the original image has been transferred into a regular grid of points.

**Notes**

In the above example we copied an image and pasted it as a table. Having the data content of an image in a table would allow us to use Manifold’s many transform operators for tables to alter the image data in a very free-form way. We could even create active columns in the table that use formulas to compute new values based on the Red, Green and Blue channel values of the image. We could then copy the table and paste it back as an image so that the image's pixels would be colored using the transformed values.

Working with image data in the form of tables will, of course, can be very slow since database tables containing a record for each pixel in an image can be very large. However, it is nice to know that in Manifold one has this option.

The example above shows a table being copied and pasted as an image. This is the normal path for creating an image from a geocoded table. The example shows a usage where a raster image might be converted into a vector data set of points, a “grid” as it is known in some systems. The fastest way to do this, of course, would be to copy the image and to paste it as a drawing.

Note that copying an image and pasting it as a drawing is not the same as vectorizing it to automatically extract points, lines and areas from the image. Pasting an image as a drawing simply takes the position of each pixel and creates a point at that location in a drawing. The drawing’s table can then be loaded with one’s choice of information (channel values, intensity, etc) extracted from the image.

**See Also**

Copy and Paste As  
Create a Map from a Geocoded Table  
Create a Geocoded Table from a Map

**Tech Tip**

At times we would like to fetch a component that’s been saved in a .map project file for use in another project file. One way to do this is to open a second instance of Manifold and to open the saved .map project. We can then Copy components from that project file, switch to our current working instance of Manifold and Paste them into our current project.
Painting into the Alpha Channel

The alpha channel of RGBA images controls the transparency of each pixel in the image. Each pixel in an RGBA image has a value from 0 to 255 in the alpha channel that specifies how transparent that pixel should be. A value of 0 means no transparency and a value of 255 means total transparency.

Just like we can turn on and off the R, G and B channels of an image we can also turn the alpha channel on and off. If all the channels of an RGBA image except the alpha channel are turned off, the alpha channel will appear in shades of gray from black to white that show the transparency that will apply in different parts of the image. Because Manifold painting tools apply only to those channels that are enabled, we can enable only the alpha channel and then paint into it using different shades of gray to "paint" different levels of transparency into different parts of the image.

This example explores use of the Paint Bucket and Gradient painting tools used with different colors and different opacity settings to create different transparency effects.

We will use our standard bronze image as an example. The window on the left shows the bronze image in an image window. We have selected all pixels outside the bronze monument and have deleted them. We then used Edit - Select Inverse to invert the selection and thus select only the pixels inside the monument. See the Selection topic for examples of selection techniques.

The window on the right shows a map window open to a map that has two layers: the bronze image on top and underneath our sample schloss image of the Neuschwanstein castle. The schloss image has been colorized to provide a softer visual background for the bronze image on top. We have used the Image - Convert To command to convert the RGB bronze image into an RGBA image.
Throughout this example we will use the Border selection style. The region of selected pixels is outlined in red. We will be working in the image window, as shown on the left. The map window as seen on the right will show the effect of each operation. Each window in Manifold has its own channel settings so we can configure the image window on the left to show only the alpha channel while the map window on the right shows the R, G, B and alpha channels. Channels may be turned on and off in the Layers pane.

We begin by turning the R, G and B channels off in the image window. This leaves only the alpha channel on. The image appears in black since all pixels have a zero alpha value, that is, zero transparency.

**Using the Paint Bucket**

We will now set up our tool controls to use the paint bucket tool.

In the Tools toolbar for images we choose the Paint Bucket tool.

In the Tool Properties pane we choose Opacity of 10.

In the Format toolbar we change the foreground color to white.
Let's consider what the above settings will accomplish. Whatever we pour the paint bucket into, we will be pouring white color with opacity of 10. That means that the white color will pour into pixels with only about 10% of the effect of pure white. If the white color is poured into a region of black pixels, they will lighten by about 10% with each pour of the paint bucket. After ten pours the color should be nearly white. Note that the paint bucket will pour paint only into the region of the image that is selected. All parts of the image outside the selection will be unaffected by the paint bucket.

If we now pour the paint bucket into the black pixels in the image window a few times we see they will lighten into gray color. With each pour of the paint bucket we lighten the gray shade that we see in the alpha channel in the image window. Simultaneously the full RGBA image we see in the map window will become more transparent and more of the schloss image will show through. A lighter shade of gray in the alpha channel indicates a higher alpha value for each pixel, or more transparency.

We can switch the foreground color in the Format toolbar to black. If we now pour the paint bucket (leaving the opacity at 10) it will darken the region of pixels with each pour.

Darkening the shade of gray by pouring darkness into the alpha channel will reduce the transparency of the image. Darker shades in the alpha channel mean less transparency. Less of the schloss image will show through.
Note that we could always have chosen some shade of gray for the foreground color and poured that shade of gray directly into the selected region in the image with opacity set to the default value of 100. Pouring pure black or pure white color with opacity of 10 is an indirect method. It has the advantage of allowing us to pour several times and to adjust the transparency interactively. This indirect, interactive method can be easier than choosing various shades of gray in the color picking dialog until we get the one that provides the transparency we want.

Also note that the alpha transparency takes on the intensity of the color painted into it. We could have selected colors other than black or white (or shades of gray) for the foreground color in the format toolbar and then painted with those colors. If a different color, such as light blue, were poured into the alpha channel it would appear in the alpha channel as the shade of gray that represents the intensity of that light blue color. Because it is difficult for most people to accurately estimate the intensities of colors other than shades of gray it is best to use shades of gray when painting into the alpha channel.

**Using the Gradient Tool**

In the Tools toolbar for images we choose the Gradient tool.

In the Tool Properties pane we choose Opacity of 100. This will allow us to paint the gradients with full opacity. Painting with the gradient tool therefore will totally override the previous colors of the pixels.

We will leave the foreground color black and the background color white. The gradient tool will create a gradient from black to white, beginning with black at the point we begin our click and drag and ending in white at the point where we end our click and drag. All parts of the image beyond the beginning will be solid black and all parts of the image beyond the end will be solid white.

Note that the gradient tool will paint only into the region of the image that is selected. All parts of the image outside the selection will be unaffected by the gradient tool.

In the image window we click where we want the gradient to start and then drag down to where we want the gradient to end and then release. The + cursor shows where we will release.
Examples

Everything before the beginning of the click and drag is painted black and then the gradient transitions from black to white at the point of release. This causes a range of smoothly increasing alpha values in the alpha channel (shown as the smooth transition in gray scale from black to white) that are seen in the map window as smoothly increasing transparency. In the map window we can see that the head and shoulders of the monument are not at all transparent while the lower right regions of the monument are completely transparent. We will leave the selection color enabled in the map window so we can see where the bronze image would be if it were not transparent.

We can click and drag again with the gradient tool to create a transition from black to white over a shorter distance.
The result of the shorter click and drag is a gradient that rapidly changes from black to white. Note that much more of the image in the map window is now fully transparent.

Note also that the pixels have not been deleted (as they might be in the case of invisible pixels) but have simply had their alpha transparency turned up to high transparency values. At any time we can paint different colors into the alpha channel to change the transparency of any of the visible pixels. For example, we could paint a darker color into the alpha channel to lower the transparency and make the pixels less transparent. Or, as seen in the steps above, we could use the gradient tool to paint different gradients into the selected region.

We can use the gradient tool in any direction. In the above screen shot we have clicked at the upper right and dragged down to the lower left.
This results in a gradient being painted into the alpha channel that is dark in the upper right and lighter in the lower left.

Although the above examples have shown the gradient tool being clicked and dragged mostly within the selected area, if desired we can click and drag it in any part of the image. Because the effect of the gradient tool will be seen only within the selected area, we have been clicking and dragging mostly within the selected area in these examples. That makes it possible for the effect of the gradient tool to be easily seen as it is applied to the visible pixels in the monument.

**Edges**

We can use almost any effect or tool we desire on the alpha channel. If we want to restore the alpha channel to no transparency (all black), the easiest way is to simply use the Brightness / Contrast command in the **Image** menu to turn the **Brightness** all the way down to zero.

This has the effect of making all the pixels in the selected region's alpha channel black. This restores the image to zero transparency for all visible pixels.

We can click on the **Selection Style** button to toggle use of red selection color on and off to show the region of selected pixels. If we temporarily turn off selection color, we can see how the image looks. Note that from the map window screen shot on the right we can see that we did a sloppy job of selecting the pixels of the monument, as can be seen by the halo of white edge pixels in the lower part of the monument.
Even if we are careful when selecting pixels within images it is often difficult to avoid some "halo" effect in parts of the image where transitions from dark pixels to light pixels occur. Such transitional regions often contain anti-aliased pixels that help provide a smoother appearance in the context of the entire image but which stand out as unnaturally light when only part of the image is visible.

We will deal with this in a subtle way. First, we turn selection color back on. We then use the Edit - Modify Selection command to create a Border selection using a border that is 2 pixels wide on either side of the previous edge. This takes the edge of existing selection and creates a narrow selection that follows the edge in the given width.

In the screen shots above, if we look closely into the map window we can see that the selection is a ribbon of border that is a few pixels wide that follows the edges of the previous selection. This new selection covers all of the white pixels that were making an ugly halo at the bottom of the monument.

If we were to zoom into the image and use the dense dots selection style to show selected regions, the illustration on the left shows a close-up view of the edge of the selected region in the monument. The view on the right shows the same view after application of the Edit - Modify Selection - Border command with a value of 2 for the border. Normally, the border created would extend two pixels to either side of the previous edge for a total border width of five pixels. Because the edge of the previous selection in this case is bordered by invisible pixels the border ends up being only three pixels wide.
We now use **Brightness / Contrast** once more to make all the pixels in the selection white by increasing the **Brightness**. Making all the pixels in the selection white makes them completely transparent. This gets rid of the halo pixels as can be seen in the map window when we turn off selection color temporarily; however, it also gives the appearance of "cropping" the pixels in the upper part of the bronze monument where there was no halo problem.

The solution to this problem is to use the gradient tool once more. We click in the middle of the image and drag down and release at the bottom. Even though we have temporarily turned off selection color the selection is still there. The gradient tool will work only on that three-pixel wide ribbon of selected pixels that follows the border of the image. The effect of the gradient will cause the upper part of the selected border to be black for zero transparency and the bottom part of the selected border to be white for total transparency.
This creates the effect we want. Note that the image in the map does not show any clipping of pixels about the head of the monument while at the feet there is total transparency so that the halo white pixels do not show because they are completely transparent. If we look very closely at the image window screen shot on the left we can see the gradient in color from black to gray to white in the thin ribbon of pixels at the border of the image.

To see the gradient effect in the border, we can use Edit - Select Inverse once more to select all pixels except those previously selected in the border region. This selects all pixels outside the thin border selection. We can then use Brightness / Contrast to increase the brightness of all these pixels to full white. This will make all pixels completely transparent except those pixels that were in the border selection. This allows us to see the gradient from black to white in the image window at left and the resulting effect at in the map window at right.

There are, of course, different ways of dealing with “halo” pixels than the above. We used the gradient and transparency because in this particular image the halo problem existed only in one region of the image.

See Also

Images and Channels
Create a Circular Feathered Image

We would sometimes like to create a circular region of an RGBA image where the region "feathers" from an opaque image to complete transparency. This procedure shows how to apply RGBA pixel transparency through desired parts of an image.

**Step 1: Import the Sample Bronze Image**

We will use the sample bronze image used throughout this documentation. Create a new project with File - New and import the image with File - Import - Image.

**Step 2: Copy the Sample Bronze Image**

Right click on bronze in the project and choose Copy. Right click in any blank region of the project pane and choose Paste As - Image. This creates a new image called Bronze 2.

**Step 3: Open the Bronze 2 Image**

Double click on Bronze 2 to open it in an image window.

**Step 4: Make a Circular Selection**

With selection mode set to Replace Selection, make a circular selection.

The most convenient way of doing this is to use the ALT - Select Circle command to select a circle on center.

**Step 5: Change the Selection to Black Color**
Use **Image - Adjust - Hue / Saturation** to reduce lightness to zero by moving the **Lightness** slider all the way to the left. Because we have a selection, the command operates only on the selection. Reducing the lightness to zero makes all the pixels in the selection black.

**Step 6: Invert the Selection**

Choose **Edit - Select Inverse** to invert the selection. This selects all the pixels *other* than those that were previously selected.

**Step 7: Change the Selection to White Color**

Use **Image - Adjust - Hue / Saturation** to increase lightness to the maximum by moving the **Lightness** slider all the way to the right. Because we have a selection, the command operates only on the selection. Increasing the lightness to the maximum makes all the pixels in the selection white.
Use **Select None** to deselect all pixels. (A one-pixel border has been added to the image above so that the image size is visible against the white background of this help window.)

**Step 8: Apply Gaussian Blur**

Use **Image - Effects - Gaussian Blur** to blur by 30 pixels. The blur will "feather" the edges of the solid black disk into the white pixels using a gradation of color from black to white. This is often used to create shadow effects, but in this case we will use it to create a circular gradient effect.

**Step 9: Convert to Grayscale Image**

We will be using this image as an alpha channel. To use it in a channel, it must be a grayscale image. We use **Image - Convert** to convert it into a grayscale image. There will be no change in the appearance since there are only black, white or gray colors in the image.

**Step 10: Open bronze Image and Convert to RGBA**
Open the bronze image by double clicking on it in the project pane. Use Image - Convert to convert it into an RGBA image. We cannot apply per-pixel transparency unless it is an RGBA image.

**Step 11: Replace the Alpha Channel**

With the focus on the bronze image, we use Edit - Load Mask/Channel to load the image’s alpha channel with Bronze 2. This creates the feathered, circular image. We may now use it as a layer within maps to compose images in multiple layers.

**Discussion**

We can generalize the above procedure to create any number of shapes in black that are then feathered with Gaussian Blur or with other methods and used to guide the alpha channels of RGBA images. Because we started with a copy of the image to be adjusted, any patterns we draw into that copy (that will be used to control the alpha channel) will automatically be lined up with the original image.

We can create unique visual effects by combining feathered images as created above with copies of the original image in maps.
For example, suppose we have made another copy of the original bronze image and then have applied Image - Adjust - Threshold to give it a black and white posterized style.

We could then create a map that has two layers: the threshold bronze image in the lower layer and the feathered circular image in the upper layer. Because the feathered upper layer has a gradient of transparency we achieve a smooth color blend from the surreal threshold image to the real color image.

See Also

Painting into the Alpha Channel
RGBa Pixel Transparency
Layer Opacity
Manually Georegister an Image

The usual way of georegistering images is to use the automated georegistration tools provided within Manifold. These tools use control points to re-scale, reposition and warp an image to match a known good drawing or image.

In cases where images need not be warped but only need to be re-scaled and repositioned we may use manual registration. This topic illustrates that procedure. It should be read not just to learn about manual methods but also to better understand how the default usage of Orthographic projection applies to images.

When importing images into Manifold we will usually import the image from some format that does not save projection information. Quite likely the image will be provided in .jpg or some other non-geographic format. Images of geographic scenes that are imported from non-geographic formats usually fall into one of four classes:

- Overhead or near-overhead photos. These may be treated for almost all purposes as Orthographic projection. The default usage of Orthographic by Manifold to import such images provides a good basis for use in a geographic context. Such images often may be georegistered simply by changing the origin and scale of the Orthographic projection used to interpret the image's data.
- Snapshots of data sets that are arranged in Latitude / Longitude projection, but which were imaged on computers or saved in formats in a way that the projection information was not captured. Many images found on Internet that show data sets such as ocean temperatures or other data fall into this class. Such images are imported into Manifold as if they were in Orthographic projection. To georegister them we first change their coordinate properties to Latitude / Longitude projection so Manifold correctly interprets their contents in a geographic context.
- Images that are irregularly scaled or warped so that they are neither Orthographic or Latitude / Longitude. Scanned images of paper maps that are in some other projection fall into this class. Such images require the use of Manifold's automated georegistration tools for use within geographic contexts.
- Less frequently, one encounters images saved in a format that does not automatically provide projection information but for which projection information is precisely known. For example, AVHRR images of the US are often encountered as images that have been massaged into Lambert Azimuthal Equal Area projection with known characteristics. Manually setting the projection parameters for the image's properties often is the only step required to georegister such images.

This example provides an instance of the second class of image: a Latitude / Longitude data set that was published as a simple .jpg image.

Our task is to take the globe.jpg sample image and to georegister it to a Latitude / Longitude drawing of world boundaries.

**Step 1: Create a drawing of world boundaries**

Next we import the World_eg.mfd sample drawing (called world in this example for brevity). We open the drawing and use Boundaries in the transform toolbar to create boundary lines on the peripheries of areas shown in the world drawing. While the newly created boundary lines are selected, we Edit - Cut them out of the world drawing. While the focus is still on the world drawing window, we use File - Create - Drawing to create a new, blank drawing in the project pane that shares the Latitude / Longitude projection of the world drawing. We rename this new drawing world_boundaries and double click it open. We can then Edit - Paste the previously cut boundary lines into the world_boundaries drawing. Press Zoom to Fit to see the lines in the drawing [the blank drawing is initially opened at such a high zoom level that we are looking into a blank spot between the newly pasted lines].
Step 2: Create the map to be used.

We create a map using world_boundaries and open the map. Drag and drop the globe image from the project pane into the map window.

The illustration above shows the situation just after we drop globe into the map window [the map is panned and zoomed so that only part of the contents are visible]. The map was created using world_boundaries so it is in Latitude / Longitude projection as is world_boundaries.

A Digression

Let's take a quick digression from our example for educational purposes.

Images imported from non-geographic formats like .jpg are brought into Manifold using Orthographic projection using one meter per pixel with the lower left corner at 0,0. When dropped into a geographic map the 0,0 coordinate is located at the intersection of the Prime Meridian (0 longitude) and the Equator (0 latitude). The image is in the map shown above but it is far too small to be visible at the zoom range shown. To show the spot where the globe image is located, we have drawn a series of red circles at the 0,0 intersection of the Equator and the Prime Meridian. These have been drawn in a layer called circles.
We can zoom farther into the 0,0 location.

If we zoom extremely far into the map at the 0,0 location we will see a dot where the globe image is located.

We don't need to zoom in to see where the image is located when it is imported by default. Understanding how images are imported by default into a projection context will help dealing with images in a more sophisticated fashion.

**Step 3: Change properties to Latitude / Longitude projection**

The native projection of the globe image is Orthographic. Although the globe image is imported into Manifold using Orthographic projection like all .jpg images, it is clear from the "unrolled cylinder" appearance of the image that it is intended to represent raster data shown in a Latitude / Longitude projection. To match the image to the world_boundaries map we need to tell Manifold to treat this as a Latitude / Longitude projection and not as an Orthographic data set.

To change the way Manifold interprets this data we open the globe image in its own window and then use the Edit - Assign Projection dialog and change the projection from Orthographic to Latitude / Longitude. We can then close the globe image window.
We leave the other parameters at their defaults for the time being. Changing coordinates properties in this way makes no change to the data inside the image. It simply tells Manifold to treat the data differently.

The result as seen above in the map window is to greatly expand the globe image relative to the size of the world_boundaries drawing. The drawing appears much larger because pixel coordinates that were once meant to be interpreted as X,Y coordinates in meters are now interpreted as X,Y coordinates in degrees. Because a degree is vastly larger than a meter, the image has expanded in size from a tiny dot a few hundred meters in width off the coast of Africa to something that is hundreds of degrees wide.

If we right click on the globe layer and choose Register, we can see the settings in use in the Register dialog. Note that X and Y unit of measure is Degree. These were the default settings for units when we changed projection in the Edit - Assign Projection dialog and they result in each pixel being treated as if it were one degree in size.

**Step 4: Re-scale the image using the Register dialog**
We need to reduce the size of the globe image by changing the scale specified by the X and Y settings for Local scale.

When rescaling images manually, we can check the Preview box and then change Local scale settings through trial and error, or we can make a first guess based on observation or information we might have about the image. If we look at the map screen shot above, it appears that the image is about three times larger than it should be. We can try to reduce the scale threefold.

If we change the Local scale values for X and Y to 0.333333 as shown above to reduce the scale to one third, the image will be reduced in size threefold. It turns out that this is a good match.

Making the above changes in the Register dialog will have the result shown above in the map window. The globe image now appears to be about the same size as the world_boundaries drawing. We can zoom in to see the alignment of the data sets better.

**Step 5: Move the image to align it with the data set.**

The screen shot above shows that the lower left corner of the image is still located at the 0,0 intersection of the Equator and the Prime Meridian. We would like to move the left edge of the image Westward 180 degrees, and we would like to move the bottom edge of the image Southward 90 degrees.
We can do this by changing the Local offset values for X and Y in the Register dialog. Right click on the globe layer tab and choose Register. With Preview on, we can change the X value for Local offset to move the image left and right horizontally. Decrease the X value for Local offset to move the image Westward. It turns out that -180 is the right value.

Seen zoomed in a bit the map shows that the image has been moved to the left 180 degrees when a value of -180 is used for X in Local offset. Although we could have found this value by trial and error, a more experienced user would have seen immediately the number was -180.

We now need to move the image Southward by adjusting the Y value for Local offset. If we want to move the image Southward 90 degrees we will need to change the Y value for Local offset to -90.
Applying these values in the Register dialog will align the globe image with the world_boundaries drawing. Closer inspection will reveal that the image is very well aligned with the drawing except for Antarctica. This is likely an error in the “projection” of the original data set for Antarctic regions.

We can see the close alignment by zooming in to Africa. If desired, we can adjust the image and drawing slightly to provide a more dramatic effect.
To create the above screen shot, we increased the contrast of the globe image slightly to provide a greater contrast between ocean areas and the land areas. We then selected the ocean areas and reduced their brightness to black. See the Selection in Images example for this sequence of steps.

We then changed the formatting in the world_boundaries layer to change the color of lins to bright yellow. We then changed Layer Opacity of the drawing layer to 60%.

Comments

Note that a very close examination of the shores of North Africa in the screen shot above might suggest that the image has been moved too far southward. If we wanted to "nudge" it slightly upwards we could use an Y offset value of -89.5 instead of -90. However, the image in use is a rather low-resolution image and probably does not merit chasing after a perfect georegistration. It's fine for providing an overview presentation.

Images we download from Internet will often be summary images and not the original data. For example, the globe image originated in a very high resolution 1 km data set that was sampled and re-sampled downward in resolution many times. The important thing is that manual registration (or better, use of the georegistration tools in a more automatic way) provides a ready means of using images in a real geographic context.

Manual georegistration of some images, like our Latitude / Longitude example, can become remarkably fast with practice, especially when similar images are to be registered. The process set forth above can be reduced to only two dialogs: one dialog to change to Latitude / Longitude projection and then one use of the Register dialog to change Local scale and Local offset. For simple images, some expert users find this faster than clicking control points and using the automated georegistration tools.

The example above is slightly contrived since we guessed the new scale factor (one third, or 0.333333) right away. In many applications we will have to choose a scale factor by trial and error to get close to the right factor. We will then shift the image through Local offset and then once again change the Local scale once the image is closer to final alignment and adjustments in scale can be better seen.

Could we have used more 3's or fewer 3's after the decimal point for Local scale, so that we could have had values like 0.33 or 0.3333333333? Yes, of course. This particular sample image is a low precision image suitable for presentation purposes and not intended for real science, so the choice of six 3's of precision was purely arbitrary.

Other Options

An equivalent way to achieve the same effect as the example above is to scale and shift the internal coordinate system of the image in one step using the Projection dialog as follows:

- Create the map as set forth in Steps 1 and 2 above.
• Open the map in a window so the effect of changes can be seen.
• Open the image in an image window.
• Use the Edit - Assign Projection dialog to change projection to Latitude / Longitude.
• Change the Local offset values to -180 for X and -90 for Y.
• Change the Local scale values to 0.333333 for both X and Y.

We can use the projection dialog when we know in advance what scale and offset is to be applied to a given image. This is useful when registering a series of similar images.

See the Edit - Change Projection topic for a discussion of the Local scale and Local offset options and their relationship to traditional scale correction, false easting and false northing values.

Another way of georegistering the image is to use control points as discussed in georegistration.

We can use the Control Points pane to place control points in the World drawing and at equivalent positions in the Globe image, we can then use the Register button in the control points pane to register the Globe image to the World drawing.

The Simple method is equivalent to changing Local offset and Local scale.

See Also
Examples

For fine adjustment of registration, use the layer repositioning commands.
Image Effects in Maps

Combining images with drawings in maps can lead to spectacular presentation graphics. Image effects allow a wide range of graphics arts effects when combined with drawings. This topic creates a presentation map using editing in drawings, drop shadows created with Gaussian blur, and a color gradient applied to an image.

**Step 1: Prepare drawing**

Import US_Main and create a drawing using the procedures given in the Create a Projected US Map topic.

Open the drawing in a drawing window. We will use Smart Mouse Selection and drawing editing to move states around. Make sure the mouse is not occupied with a command mode (such as Zoom or some other mode) and click on the state of Washington.

The state becomes selected as the primary selected object, ready for editing, with edit handles appearing. Edit handles overlap densely at the zoom level illustrated. Click on any of the edit handles and drag to the upper left.
Clicking and dragging an edit handle moves the entire object to the upper left.

Release the click and drag and the state of Washington snaps to its new location.

Click on the state of Oregon to select it for editing and then click on any edit handle and drag it to the left.
Release the click and drag and Oregon will move to its new location.

Continue in this way to move western states away from the main part of the drawing. Save the project as a .map. It is always wise to save work in a project against the chance of irreversible user errors.

**Step 2: Create a Drop Shadow**

We will now create a "drop shadow" for this rendering of the United States. A drop shadow is a graphics layer that gives the appearance of a shadow, as if the item above it were physically suspended above the page and casting a shadow. To create a drop shadow we begin by saving the drawing as an image.

Choose Tools - Make Image command, saving the image using the name "Shadow Mask." We've picked a name that sound a bit strange for now, but it will make sense later.

The default size of the image will be that seen for the drawing at display monitor resolution. Choosing the image size at this stage of our procedure will affect the quality (resolution) of the drop shadow. If the map we are creating will be printed out on a large scale we may wish to use a much higher resolution image. To do so we could first create a print layout for the drawing and then use the Make Image command for the layout (when used with layouts the Make Image command allows specification of image resolution).
The result is an image called **Shadow Mask**. We open this image in an image window and see it is an image that is 1200 by 900 pixels in size that is a raster image rendering of the drawing we had created. It appears with blurry border lines in the illustration above because the view is zoomed out to enable the image to be seen at reasonable size in this topic.

We have zoomed out far enough so the extent of the image is clear. Beyond the edge of the image we see the default, "transparent" background. Note that when Manifold created the image from the drawing it used the background color, white, to fill in the "empty" white space around the area objects.

We will use this image to create a drop shadow. However, because the blue regions from the area objects press right up to the edges of the image we first need to add margin to the image to provide a larger image into which the Gaussian blur can reach.

We add margin using the Transform toolbar and the Add Margin operator. As seen above, we will add 100 pixels to the margin.
The result is a 100 pixel margin added in black color. To make the black margin match the rest of the image we will use the Paint Bucket tool to fill it with white color.

Click on the foreground color in the format toolbar for images and make it white (the illustration shows both foreground and background color wells in white color).

Click on the paint bucket tool and then click anywhere in the black margin.
The paint bucket tool will fill the region of contiguous black pixels with white color.

Our next task is to make all pixels in the blue map region black in color. We need to select all pixels that are not white.

The easiest way to do this is with Select Touch and SHIFT clicking onto any of the white pixels. This selects all of the white pixels in the image. We use SHIFT to extend the touch selection to all white pixels and not just those that are contiguous to the pixel on which we clicked. Had we not used SHIFT there might have been some white pixels in the Great Lakes or the upper parts of Chesapeake Bay or other enclosed regions that would not have been selected.
To select all non-white pixels we simply choose Edit - Select Inverse to invert the selection. It would have been difficult to select all non-white pixels using touch selection because they are a mixture of blue and gray colors. In this case it was much easier to select the white pixels and the invert the selection.

We launch Hue / Saturation and move the Lightness slider all the way to the left to force all of the selected pixels to black. Press OK.

The result (seen after pressing Select None so there is no selection) is that all of the non-white pixels have been converted to black color.

We can now create a Gaussian blur shadow by choosing Gaussian Blur. The number of pixels used for the blur depends upon the shadow effect desired and the resolution (size of the image in pixels) of the image. Using larger numbers of pixels results in larger "shadows." Large and faint shadows can provide a visual cue that the object is "floating" farther away from its background. Smaller and dark shadows indicate the object is closer to its background.

Beginners might want to experiment with Preview checked to see the effect obtained.
The result is a blurred image of the black pixel regions. The astute reader may have already thought ahead to how this might be used as a layer under a vector drawing layer to create the impression of a shadow.

**Step 3: Create a map using a shadow image**

To use a shadow image under a drawing layer we need to work with a map, because only maps in Manifold have layer structure as we would like to use. We use the US_Main drawing to create a map. We then open the map. Because we will need the Shadow Mask image for other purposes, we will first Copy the Shadow Mask image in the project pane and Paste to create a second image that we call Shadow.

We drag and drop the Shadow image into the map, right click on the Shadow tab and choose Order to move it below the US_Main layer. The illustration above shows the tab separately from the illustration of the map to
provide a more compact illustration. Note that the image is correctly georegistered to the drawing. It was originally created from the drawing and so continues to be georegistered to the drawing. However, the blur effect is not what we would like. To create a drop shadow effect the image should be displaced from the drawing.

To offset the drawing from the shadow we click on the **US_Main** layer to make it active.

Using the **Grabber** tool, we do a CTRL-click-and-drag slightly to the upper left. This moves the drawing relative to the drop shadow. We could have clicked on the **Shadow** layer and used CTRL-click and drag slightly down and to the right to move the shadow instead of moving the drawing. The visual effect obtained would be the same. Note: moving layers with a CTRL-grabber works only if the layer and the map both use the same projections, differing only in local offset or local scales.

At this point, the drop shadow effect is achieved for many purposes. If desired, we can right click on the **Shadow** layer and choose **Opacity** to alter the opacity of the shadow layer. Altering the opacity of a layer bearing a drop shadow will alter the lightness of the shadow and can provide a more realistic effect.
However, there is a problem with the map we have constructed that unchecking the Background box in the Layers pane reveals. With the background turned off we can see all layers against the checkerboard default background. In particular, we can see that the Shadow image has a large region of white pixels. The shadow is constructed in that image by blending black pixels in a Gaussian blur into a field of white pixels. That's fine for an effect if we desire a white background and if the Shadow layer is to be the lowest layer in our map.

**Step 4: Create a drop shadow using alpha transparency**

Suppose, though, we would like another drawing layer to appear below the US_Main layer and its shadow, so that the United States appear to float above another drawing and to cast a shadow on that layer? In that case, the shadow layer must be constructed so that the black shadow blurs or “feathers” into transparency, not into white pixels. This is an effect that can be achieved using a short sequence of sophisticated steps.

First, we open the Shadow Mask image in its own window. We then use Image - Convert To to convert it into a Grayscale image. We then close the Shadow Mask image.

Next, we open the Shadow image in its own window. We use Image - Convert To to convert it into an RGBA image.

With the focus on the Shadow image window, we launch Hue / Saturation and move the Lightness slider all the way to the left to force all of the pixels in the image to black. We leave this image window open because we have one more command to apply to the image.
We can see the effect of forcing all of the pixels in the **Shadow** image to black. It seems very counterproductive since it destroys the shadow effect.

With the focus on the **Shadow** image window we choose Edit - Load Mask/Channel.

![Load Mask/Channel dialog](image)

In the **Load Mask/Channel** dialog we load the **alpha** channel using the **Shadow Mask** image. This is why we named it "shadow mask" when it was created and why we converted it to a grayscale image. Only grayscale images can be used as masks. Because we created the **Shadow** image from the **Shadow Mask** image both are exactly the same size and thus the **Shadow Mask** can be used as a mask. Remember that the **Shadow Mask** image shows a Gaussian blur drop shadow in shades from black to white. If we use this as a mask for the alpha (transparency) channel of an **RGBA** image we will be superimposing a transparency gradient upon the **Shadow** image that follows exactly the pattern of the Gaussian blur in the **Shadow Mask** image. Applying such a transparency gradient to the all black **Shadow** image will result in a gradient from solid black pixels to fully transparent pixels with partial transparency in the region of the Gaussian blur effect. We will have achieved a blur from black into transparency.
The moment we press OK in the Load Mask / Channel dialog we can see the effect in the map window. There is no longer any solid appearance to the Shadow layer in any location other than the actual shadow. We can now create layers under the Shadow layer and they will show through as desired.

**Step 5: Add a graticule**

To show the effect of the transparent drop shadow we can add a graticule underneath the US_Main and Shadow layers.

Right click the US_Main layer tab in the drawing and choose Add - Drawing to add a drawing. Name it Graticule.

In the Tools - Options - Colors dialog change the graticule color to black.

Choose View - Graticule to launch the Graticule dialog. Use the settings shown above and push Create to create a graticule as line objects in the Graticule drawing layer.
The result is a graticule grid that appears in the layer. We can right click on the layer tab and choose **Order** to move the layer tab to the bottom of the layer stack.

To achieve a lighter shadow effect, we right click on the **Shadow** layer and choose **Opacity**. In the **Opacity** dialog we set the transparency of the layer to **50** percent and press **OK**.

The result is a map using a transparent drop shadow that allows layers beneath it to show through. Note how the partially transparent shadow produces a natural effect where it overlays the graticule lines.

**Step 6: Add a gradient color effect**
A combination of images and drawings in maps can create many effects in addition to drop shadows. For example, we can color the US using a gradient of color. [This example shows a purely artistic effect. If we want to show data as gradients we could use surfaces.]

To create the effect desired we need to create another image from US\_main. We do this by opening the US\_Main drawing in its own drawing window and then choosing Tools - Make Image as we did before.

We will create an image called Gradient.

If we click open the Gradient image in its own window we can see that it is similar to the Shadow Mask image we created to use in building a drop shadow.
Using the **Select Touch** tool we **SHIFT**-click on the white pixels to select all white pixels.

Press **Delete** to delete the white pixels. We now need to select the other pixels. To do this, we **SHIFT**-click onto the region of invisible pixels and then choose **Edit - Select Inverse** to invert the selection. The result is to select all of the remaining, visible pixels.

To create a color gradient we choose the colors in the format toolbar foreground and background colors that will define the gradient.

Select the gradient tool.
Create a gradient by clicking on the left side of the image with the gradient tool and then dragging to the right side and releasing. The color gradient will be created along this line within the selected pixels.

Instantly, we achieve a dramatic coloring effect in the Gradient image.
If we drag and drop the Gradient image into our map as the uppermost layer we see the dramatic effect in the map as well. Because the Gradient image was created from the US_Main drawing it is georegistered to that drawing and overlays it perfectly.

We may want to show borders for states in the composite map. To do this we must create a drawing layer of state boundaries above the Gradient layer.

Click on the US_Main layer and run the transform toolbar using the Boundaries operator.

This creates line objects in the US_Main layer that outline the boundaries of the state areas. We can cut these out of the drawing using CTRL-X or Edit - Cut. We can then right click onto the Gradient layer and choose Add - Drawing to create a new drawing. We can then paste the boundaries into the new layer with CTRL-V or Edit - Paste.
The final result is a dramatic map that was created using a mixture of image and drawing techniques.

**Comments**

The images used in this procedure are slightly larger than necessary because the initial size of **1200 x 800** used in the Make Image dialog results in unnecessarily large borders around parts of the image. We could have reduced the size of the images by using Crop after creating the Gaussian blur.

It is usually better to create a drop shadow effect by repositioning the layer containing the drop shadow than it is to move the objects being shadowed as was done in this example.

The map shown uses a gradient image at the end to color the states of the US. If desired, this step could be skipped and the "blue" drawing thematically formatted. Moving the states to new positions for artistic effect does not change their status as objects: they can still be thematically formatted to show trends in population or other variables.

The planned use of the map should be considered when specifying the size of the drop shadow created using Gaussian blur. A blur of **30** pixels as used in this example works well when a large image is seen in a zoomed out view as a small portion of the monitor. Using a blur much smaller than that would result in a too-thin shadow. However, when zooming into the view or printing on a larger sheet of paper 30 pixels of blur may prove too much. Experience will reduce the need for trial and error.

While gaining experience, make a copy of the black and white image and then apply Gaussian blur to the copy. One can then relatively easily fall back and use a different level of blur if the original effect proves disappointing. Changing the perceived blur in the shadow used in the map is easy, since once we create a new Gaussian blur mask we simply re-load the alpha channel with the new mask.
Create a Globe Image

In this example we create the image seen above. The task is to take a georeferenced globe image, re-project it into Orthographic projection and then add haze near the edges.

**Step 1: Prepare the base image**

We begin by using the methods described in Manually Georegistering an Image to georegister the globe image. We alter the contrast as described in Selection in Images and apply Relief to better highlight mountainous areas.

Next, we show the image in a map and re-project the map into Orthographic projection using **Edit - Assign Projection**. Click the **Suggest** button for reasonable values in the map projection. The map shows a georegistered image seen within a projected map. This is great for applying projections and other geographic effects but it involves a lot of computation overhead if we are doing only graphics arts work and don't care about projections or preserving geographic context.

Note: if we wanted to re-project the original image, we would use **Edit - Change Projection**. In that case, it is very important to use the **suggest** button to get sensible values for local scale, which should be the same.

Because we are interested in using this image for purely graphics arts work, we will take a moment to save it as an image. To do this, we use the **Tools - Make Image** command to save the component as image at screen resolution.
Examples

Clicked open in an image window the result is seen above. It is a nice image but lacks a sense of presence because there is no haze that appears near the edges. We will use the earth image in a map together with other image layers to create a more artistic effect.

**Step 2: Create the haze layer mask**

We will be working in a map window, so we create a map using the earth image. We will call the map earth_from_space. Next, we need to create the image layers that will modify the appearance of the earth layer. We begin with a haze layer.

To create the haze layer, we will first create a mask that will be used to apply variable transparency.

In the project pane, right click onto the earth image and choose Copy. Right click elsewhere in the project pane and choose Paste. This makes a copy of the earth image. Name this copy hazemask. Double-click open the hazemask image.

Touch select the white areas in the outer part of the image.

Next, choose Edit - Select Inverse to invert the selection. This is a quick way of selecting just those pixels that make up the planet in the image. We use Image - Adjust - Hue / Saturation to reduce Lightness all the way to the left. This makes all the pixels in the planet region black.
We now apply Gaussian Blur to blur the edges. The image above shows the effect after we have applied it and have chosen Select None to clear the selection. Note that since only the region of pixels showing the planet was selected the Gaussian blur effect is blurry inward only. [A nuance: even though the Gaussian Blur effect works within the selected region only, it computes its effect globally based on pixels outside of the selection as well. The image should be large enough so that the region of white pixels outside the selected area is reasonably large all the way around the planet’s disk. If the image is cropped so there are few white pixels except in the corners of the region, then there will be a visible difference in the Gaussian blur effect along the edges where there are fewer white pixels outside the selection.]

We then apply Invert to invert black and white values. We use Image - Convert To once more to convert the image to a Grayscale image (because masks must be grayscale). We can now use hazemask as a mask.

**Step 3: Create a haze layer image and apply the mask**

Once more we copy the earth image in the project pane and paste it as a new image in the project pane. Name this new image haze.

Open the haze image in an image window. So far, it is a copy of the earth image. Use Image - Adjust - Hue / Saturation to increase the Lightness to full. This makes the entire images white pixels.

Now we used Edit - Load Mask/Channel to load the alpha channel from hazemask. The result in the haze image is shown above. The central part of the haze image is completely transparent and the outer part is completely opaque white. The transition region from completely transparent to completely white is that part of the hazemask image that was blurred using Gaussian blur.
We can now drag and drop the `haze` image into our `earth_from_space` map as a layer above the `earth` layer. The result is seen above. The transparent central parts of the `haze` image allow the `earth` image underneath to be visible. The gradient from transparent to white creates a haze effect at the edges of the planet region.

**Step 4: Create a "background" layer**

If we want a white surrounding color we are done. However, if we wish to have a different color surrounding the "planet" we will need to add one more layer. This new layer will overlay the layers already created, but because it will be precisely shaped with a "hole" in the center for the planet it will appear to be a background.

We do this by going back to the project pane and once more copying the `earth` image and pasting it as a new image, which we shall call `space`. So far, `space` is an exact copy of the `earth` image.

We open the `space` image and touch select the white areas in the outer part of the image.
Next, we choose Edit - Select Inverse to invert the selection. We want to select a region just one or two pixels smaller than the "planet", so we use Edit - Modify Selection - Contract and contract the selection by 1 pixel. [This is a matter of taste. Under some circumstances we might use 2 pixels.] We then choose Edit - Delete to remove the pixels in the selection by converting them to invisible pixels.

Next, we use Image - Adjust - Hue / Saturation to reduce Lightness all the way so that all the pixels become black. The result is illustrated above. Finally, we apply Image - Effects - Gaussian Blur set to blur 2 or so pixels. The result is a black image with a circular region that is a pixel smaller than the planet region with a slight blur right at the edge. We drag and drop the space image thus created from the project pane into the earth_from_space image.
In the earth_from_space map window we can now right click on the space layer tab and set the layer’s Layer Opacity to achieve the color we like for what appears to be the background. Layer opacity in the 80% to 90% range results in effects as seen above.

Notes

One of the main techniques shown here is copying an original image and pasting it as a series of images that are then in turn transformed into masks and other layers. Using the original image and then creating the masks from copies of that image guarantees that the masks and space layer will be exactly the right size to use as we desire in layers in combination with the original image.

Why did we go to the effort of contracting the selection and applying a Gaussian blur when creating the space image? If we were to simply select the planet region and make it invisible in the space layer we would end up with the "jaggies" when the space layer was placed above the haze and earth layers. The harsh transition between invisible pixels and solid black pixels in the space layer would result in a jagged edge. By making the "hole" a pixel or two smaller in size than the planet disk and applying a Gaussian blur we end up with a slight transition from invisible pixels to solid black pixels. The effect is something of a manually created anti-aliasing that gets rid of the jaggies. Using a blur in a zone a few pixels wide is a good way of making transitions in images.

Strictly speaking, only the haze and space images need to be RGBA images. The earth image could be an RGB image.

It's important to understand that except for Step 1 this entire topic is about graphics arts work using images in a non-geographic way. Only in the first step do we use the images in a geographic context to create a view of the Earth in Orthographic projection. We then make an image that is a screen shot of that view. After that, all work is done on the screen shot image and not with the original georegistered image / map.

We could have combined drawings with the image in the original map from which the original screen shot was made. In that case, the earth image with which we started could have included visual effects from drawing, like the last image in the Manually Georegister an Image topic.

Just for fun, we can go back to the original geographic map and create an Orthographic projection that is centered on 0 longitude and 90 latitude for a polar aspect view. We can then make a screen shot image and process that image as noted above to create a polar aspect view from space, with haze.
Import a Shapefile

"Shapefiles" are GIS files in ESRI's .shp format. In this example we find a map on the Internet, download it, import it into Manifold as a drawing and then correctly set the drawing's coordinate properties. We will import a map of the 109th Congress districts from the USGS's National Atlas of the United States.

When importing shapefiles, if the shapefile is intended to use the default WGS84 datum and is also in latitude / longitude unprojected form then import is easy: we simply import the file using the File - Import - Drawing dialog and we're done. This example uses a slightly more sophisticated example, where the shapefile is in latitude / longitude unprojected form but using a datum other than WGS84. In this case, we do the simple import but then after import we must correct the datum from WGS84 (the default) to the desired datum.

All URLs given below were current at the time this topic was written. Since web pages change continually it is quite likely the pages given have changed.

Step 1: Locate and download the shapefiles

Visit the National Atlas Map Layers Warehouse page at http://www.nationalatlas.gov/atlasftp.html, and drill down into the site to find the 109th Congress districts layer.

We download the cgd109p020.tar.gz shapefile. Note that the data set is available in both shapefile format and SDTS format. The SDTS format is more precise because it automatically captures projection information that shapefile format does not save. However, since this example illustrates how to get around various hassles involved in using shapefiles we will go ahead and download the shapefile version.

Step 2: Find and download accompanying documentation

As experienced hunters of shapefiles, we know that we should never download a shapefile from a website without also trying to locate and download any accompanying documentation. We see that the September 2005 link appears in a column that seems so have something to do with metadata, so we click on the link. It leads to a page with lots of information including links to "spatial" metadata, which is usually where information about projections is located. After hunting around a bit we find:

b. What coordinate system is used to represent geographic features?

Horizontal positions are specified in geographic coordinates, that is, latitude and longitude. Latitudes are given to the nearest 0.0000278. Longitudes are given to the nearest 0.000278. Latitude and longitude values are specified in Decimal degrees.

The horizontal datum used is North American Datum of 1983.
The ellipsoid used is GRS1980.
The semi-major axis of the ellipsoid used is 6378137.
The flattening of the ellipsoid used is 1/298.257222.

Because the geographic coordinates are in decimal degrees we know that the shapefile is in unprojected, Latitude / Longitude form. All we need now is the name of the Datum, which we can see is the North American Datum of 1983 using the GRS1980 ellipsoid.

Many people use "datum" as a synonym for "ellipsoid" although the two are not exactly the same. Usually a named datum such as the North American Datum of 1983 automatically implies the use of a particular ellipsoid so that to specify both is somewhat redundant. In general, use the name of the datum if you can find it in the Manifold projection dialogs. If you can't use the name of the ellipsoid if you can find that. Note that there is some variation in how the names of datums and ellipsoids are abbreviated. For example, people write this ellipsoid's abbreviation as either "GRS1980" or "GRS80".
Examples

Because this example deals with the simple mechanics of importing a shapefile we state that the only data we need from the documentation is the projection information (types of coordinates / projection and datum and/or ellipsoid). The documentation also includes much other useful information such as the provenance of the data, accuracy and so on that could be very important later should we wish to document the sources and nature of our data.

**Step 3: Decompress / De-archive the download file**

The actual file downloaded is called `cgd109p020.tar.gz`. In the usual Federal way, this file employs a compression and archiving technique that is not used by 95% of computing humanity. It uses GNU-zip ("gz") compression applied to a UNIX tape archive ("tar") file. Lucky for us, mighty WinZip can deal with such weirdness.

**Very Important**: Before using WinZip to extract any files with a "tar" in the extension, make sure that WinZip’s Options - Configuration - Miscellaneous tab has the TAR file smart CR/LF conversion box NOT checked. This box is there so WinZip can convert UNIX style text files to DOS style text files if they are archived as tar archives; however, if WinZip applies the same conversion to binary files (like shapefiles or other GIS binary files) saved in tar archives it will ruin them.

![File List]

After using WinZip to extract the files we can see we have created four new files (seen above in Windows Explorer): a `.txt` file and the three files that make up the "shapefile": `.shp`, `.dbf` and `.shx` files. It turns out the `.txt` file contains a copy of the metadata we saw in the web site link. USGS has thoughtfully included this with the shapefiles because they know that shapefiles do not save projection information. However, we did not know that such a text file accompanied the shapefiles so it was wise of us in the previous step to hunt down the documentation for the shapefile, just in case. Most shapefiles found on the web will consist simply of the `.shp`, `.dbf` and `.shx` files that are part of the actual shapefile format.

**Step 4: Import into Manifold**

Launch Manifold and use File - Import - Drawing to import the `cgd109p020.shp` file as a drawing.
In the **Import** dialog use the **SHP Files** setting in the **Files of type** box.

In the next dialog, we see all the fields available for import from this shapefile. We use the default with all fields checked and press **OK**.
A new drawing appears in the project pane, complete with a table.

The districts show the exquisitely drawn, bizarre shapes designed to aggregate voting majorities of the party in power in a state and to deny the opposing party the opportunity to ever form a majority. If we clicked open the table associated with this drawing we would see that for each area object there is a record providing various attributes for that congressional district such as the name of the representative and other information.

Note that the information bar advises us the projection has not yet been verified and invites us to click on the bar to do so.

**Step 5: Provide corrected datum**

It would appear at first glance that the drawing imported correctly. If we pass the mouse over the drawing we see apparently correct latitude and longitude figures for the mouse's position. The shape of the United States has the familiar rectangular appearance of the US seen in unprojected Latitude / Longitude form. If all we will use this drawing for is thematic formatting to present information (such as a thematic overview of which districts are Republican and which are Democrat) we could use this map "as is."

However, because the default datum used when importing shapefiles is **WGS84** and not the North American Datum of 1983 for maximum precision we should change the datum to that actually intended for the shapefile used. This will give as precise as possible a spatial match between this drawing and any other drawings we might use with it in a map.

To change the datum we click on the information bar to open the Edit - Assign Projection dialog.
The datum box is loaded with the default datum used on shapefile import.

To change this, we scroll through the choice of datums conveniently organized into one massive list of seemingly endless datums until we reach the North American 1983 (mean for CONUS) datum. In a rare lapse of precision, USGS refers to the "North American Datum of 1983" in the metadata text file for the congressional districts shapefile we downloaded. However, there are in fact several different North American datums for 1983. We choose the one for "CONUS" (Contiguous US). We don't know for sure this is the datum intended by USGS, but it is the best bet given the information we have and considering that most of the congressional districts are located within "CONUS".

We press OK to apply this projection interpretation to the coordinate properties of this drawing.

Except for the disappearance of the info bar advising us to verify the projection, there is no change in the visual appearance of the drawing since any change caused by a difference between the World Geodetic datum used by default and the North American datum we specified is completely invisible. In fact, it is so small it is likely to be less than the intrinsic accuracy of the shapefile as drawn by USGS.

Note see the Import Drawing - SHP, Shapefiles topic for an example of importing a projected shapefile where more than just the datum is changed. Also, see the Import a Projected Shapefile Example.

At this point we have accomplished what we set out to do, importing a shapefile into Manifold. The rest of this example shows some elementary, optional steps illustrating typical tasks such as thematic formatting and re-projection.

**Step 6: Apply thematic formatting**

Our drawing looks boring in gray and does not convey all of the information that is in the drawing's table. We can show information in the table by using thematic formatting to color the areas in the drawing based upon the data associated with each area.

Let's have fun and color the drawing by political affiliation. We click on the background color well for areas in the Format Toolbar and choose Theme... in the pull-down menu of colors.
We set up the **Format** dialog as shown above. We changed the colors in the color wells by clicking on them and choosing a desired color. The party affiliation codes are either **D** or **R** for Democrat or Republican and **I** for Independent. To avoid hackneyed stereotypes, we’ve chosen colors other than the “blue state” and “red state” convention often used in the US.

Alert readers will notice that the number of representatives adds up to a higher number than there are seats in Congress. The numbers are higher because this is the number of areas in the map coded for each party: some congressional districts (such as those that contain islands or water areas) are represented by multiple area objects.

The drawing now shows congressional district colored by party affiliation. Some regions in gray are areas over water in the Great Lakes and not assigned to a particular district.
If we zoom out, we can see how congressional districts of the two parties are distributed across the United States.

**Step 6: Project the drawing**

Zoomed out to see the entire lower 48 states of the United States the drawing has an unpleasant, "flattened" appearance caused by the use of *Latitude / Longitude* coordinates. We can give the drawing a more pleasant appearance by either displaying it within a map that uses a projection or by re-projecting the drawing into some appropriate projection. This is a purely optional step motivated by nothing except a desire to have a more appealing visual appearance.

We can project the drawing using the Edit - Change Projection command and choose a projection desired, such as the Lambert Conformal Conic projection using *Center* numbers that are a position roughly in the center of the US.

This dialog is essentially identical to the projection dialog used within Edit - Assign Projection. However, whereas the Edit - Assign Projection version does not change the data (it simply tells Manifold how to interpret the coordinates that exist), the Edit - Change Projection dialog will re-compute the data in the drawing to new coordinate numbers.
Note that we manually entered the -96 value for the Center Longitude. We did not use the Suggest button (which would be the normal action), because the illustration we have been showing is zoomed into a view that shows only the continental United States; however, the actual drawing contains the entire United States, which includes territory spanning the globe. Alaska, for example includes islands on both sides of the +/-180 longitude line. Pressing the Suggest button would therefore suggest a value of 0 for the Center Longitude because this particular drawing spans the globe.

The result of the re-projection is a classic projected view of the contiguous, lower 48 states of the US. Note that the illustration above is a zoomed in view. Using a conic projection will result in considerable distortion within American Samoa and other regions that are far away from the lower 48 states. However, since our illustration above shows only the lower 48 states we have chosen to use a projection that looks good for that region.

Note also that we chose the "single parallel" version of the Lambert Conformal Conic. This is the easiest to specify because all we need specify are the center latitude and center longitude. It is not as low in distortion as the full Lambert Conformal Conic but for a thematic map of this size we are not worried about distortion. We would rather avoid the (slightly) extra work of specifying two additional parallels to be used.

Regional Settings

When importing data from text file formats like .csv or .dbf or from shapefiles (because shapefiles use .dbf format as part of the shapefiles) the Regional Options in our Windows system have to match settings within the file that is being imported. This is because such simple formats are too stupid to understand that in different countries people use different symbols to denote decimal points (that is, either a dot character or a comma). To change Regional Options in Windows, do the following:

- Go to the Control Panel and open the Regional Options applet.
- Go to the Numbers page,
- Ensure that the Decimal Symbol is set to '.' (dot),
- Ensure that the List Separator Symbol is set to ',' (comma),
- Press Apply to apply changes if there were any,
- Import the .csv file or shapefile desired.
- After import, restore the original settings of Decimal Symbol and List Separator Symbol if you like.

Another option is to invoke the Data Sources applet (hidden within Administrative Tools folder of the Control Panel if you're running Win2K or XP) and create a file DSN for the .csv file you want to import. After creating the DSN configure it to use the comma character as a data separator.

See Also
Import Drawing - SHP, Shapefiles for discussion of shapefile import, including the role of database drivers. This topic also provides an example of importing a \textit{projected} shapefile, including use of the Edit - Assign Projection dialog after the import to adjust projection parameters.

See also the Import a Projected Shapefile topic for a step by step example showing how to import a shapefile that contains projected data.
Import a Projected Shapefile

Because ESRI .shp format files ("shapefiles") do not store projection information they are most often used to save information in unprojected form where the contents are coded in coordinates using values that represent degrees latitude and longitude. On occasion we may encounter shapefiles that encode projected information. In such cases the contents of the shapefiles will be coded in coordinates using values that represent meters or feet in some projected coordinate system.

To use such files we must know the projection and projection parameters used to encode the shapefile. Because the shapefiles themselves do not contain this information we must receive it through other means such as a "readme.txt" file that accompanies the shapefiles or other documentation.

This example shows how to import a projected shapefile where we know what projection was used to encode the shapefiles. This example assumes a text file accompanies the shapefile to tell us what projection it is in.

**Step 1: Determine correct projection**

Our first task is to find out what projection the shapefile is in.

In Windows Explorer we can see that the "shapefile" is actually three files: a mexico.shp file, a mexico.dbf file and a mexico.shx file. In addition, we see that there is a readme.txt file that accompanies the shapefiles. If we are lucky, this text file contains information about the shapefiles such as the projection this shapefile is to use.

We can either pop open the text file in Notepad, or, using File - Import - Comments import it into Manifold as a comments component as seen above.

If we open the comments component we see the readme.txt file tells us that the projected shapefile was projected using Orthographic projection centered at latitude 24 degrees longitude -102 using WGS84.

These are "magic numbers" in that if the author of the drawing did not provide them to us we would have no way of knowing from the shapefile itself which projection and parameters to use.
Experienced user may object that this example begins right away with a page out of fantasy-land, in that it is rare that someone who is so thoughtless as to publish projected information in shapefiles will have the courtesy and foresight to provide a "readme" file that tells us what projection to use. But that happens surprisingly often, or, if the information is not in an accompanying text file it will be available in some "metadata" file or in some web page associated with the site from which we downloaded the shapefile.

People who use legacy formats normally get street-smart very quickly about the need to do a little detective work to find out what projections are intended if projected data is in play. For the sake of this topic we assume that such detective work has found the necessary information in the form of an accompanying text file.

**Step 2: Import shapefile using defaults**

We will import a projected shapefile that shows Mexico. This example is not accompanied by a sample file on the Manifold CD.

![Import Dialog](image)

In Manifold, choose **File - Import - Drawing**. In the **Import** dialog browse over to the directory in which the shapefile is located. Choose **SHP Files (*.shp)** in the **Files of type** box and open the **mexico.shp** file.
Manifold will automatically open the .shp file as well as the other two files normally used in SHP format. In the Import SHP File dialog we see this drawing has but one field, a column called Place_name. We press OK to import the drawing.

The result is a new drawing in the project pane. We can open the drawing by double-clicking on it in the project pane.
When opened the drawing appears OK. If one does not have a practiced cartographic eye one might not notice that the shape of Mexico is characteristic of projected maps as opposed to the "wide" appearance of countries in these latitudes as seen in unprojected Latitude / Longitude views. If we move the cursor over the drawing the coordinates seen in the status bar will show there is something not yet right about the coordinates in this drawing.

We can see that even though the drawing has been imported using **Latitude / Longitude** degree coordinates the coordinates shown in the status bar as the mouse moves over the drawing are values like 99517 and 56020, which are impossible values for degrees. Since shapefiles do not provide projection information but are mostly used for unprojected drawings, Manifold uses a default of **Latitude / Longitude** to import shapefiles. For correct use within Manifold, the correct projection **must** be specified for this drawing.

### Step 3: Specify correct projection

To specify the correct projection we launch the Edit - Assign Projection dialog and change parameters as necessary.

![Assign Projection dialog](image)

This is a familiar dialog used within Manifold for several similar purposes. In this case the purpose is to specify what projection parameters should be used to interpret the coordinate values in the drawing. The dialog shows the current interpretation applied, that the drawing is in **Latitude / Longitude** projection using degrees as the unit of measure for coordinates.
We expand the Azimuthal heading and choose the Orthographic projection.
We specify 24 as the Center Latitude and -102 as the Center Longitude. The datum is already WGS84 so there is no need to change the datum. Press OK.
There is no change in the appearance of the drawing. However, if we move the mouse over the drawing we can see there is a change in the status bar readouts.

| Orthographic | 101°36.336' W 24°9.028' N |

The status bar now reports the drawing is in Orthographic projection. The position of the cursor is reported in values that make sense for longitudes and latitudes in Mexico.

**Regional Settings**

When importing data from text file formats like .csv or from .dbf or shapefiles the Regional Options in the system have to match settings within the file that is being imported. This is because such simple formats are too stupid to understand that in different countries people use different symbols to denote decimal points (that is, either a dot character or a comma). To change Regional Options, do the following:

- Go to the Control Panel and open the Regional Options applet.
- Go to the Numbers page,
- Ensure that the Decimal Symbol is set to ‘.’ (dot),
- Ensure that the List Separator Symbol is set to ‘,’ (comma),
- Press Apply to apply changes if there were any,
- Import the .csv file or shapefile desired.
- After import, restore the original settings of Decimal Symbol and List Separator Symbol if you like.

Another option is to invoke the Data Sources applet (hidden within Administrative Tools folder of the Control Panel if you’re running Win2K) and create a file DSN for the .csv file you want to import. After creating the DSN configure it to use the comma character as a data separator.

**Notes**

A recent addition to shapefiles is the use of a .prj file that saves projection information. However, this is a recent addition that is not found with most projected shapefiles that are available. A further problem is that .prj files do not provide unambiguous projection information and so place users at risk of importing shapefiles that use .prj and still getting inaccurate projection information. If we are using projected shapefiles that are accompanied with .prj files, we should still make every effort to find out what projections are intended and then confirm the result of the import as in the example above.

A sample file for this example is not provided on the Manifold CD because of the risk of confusing users with a projected shapefile should someone try to open the file without reading this example first.
Import a Raw Binary File - NLCD

This topic uses the Import Surface - Raw Binary importer to import a raster data set that represents land used data in the state of Delaware in the US.

Derived from the early to mid-1990s Landsat Thematic Mapper satellite data, the National Land Cover Data (NLCD) is a 21-class land cover classification scheme applied consistently over the United States. The spatial resolution of the data is 30 meters and mapped in the Albers Conic Equal Area projection, NAD 83. The NLCD are provided on a state-by-state basis. The state data sets were cut out from larger "regional" data sets that are mosaics of Landsat TM scenes. At this time, all of the NLCD state files are available for free download as 8-bit binary files and some states are also available on CD-ROM as a Geo-TIFF.

Although there are now other land cover data sets that may be freely downloaded, this particular data set can be used to show import of binary files.

Step 1: Acquire the Data

Visit the NLCD download page currently at


and drill down into the final publication download site at

http://edcftp.cr.usgs.gov/pub/data/landcover/states/

Download the files for Delaware. Note that as always, URLs in this Help file may be changed after Help has been published. The USGS land cover web pages seem full of dead links and ongoing changes, so finding the data for this example may be difficult. The Delaware NLCD folder in the help examples folder on the Manifold CD contains a copy of these files.

There are three files to download: the data file in .gz compressed form and two text files:

- delaware_nlcd.bin.gz
- delaware_FGDC.txt
- delaware_readme.txt

Decompress the .gz file using WinZip or some other decompression package into a .bin file:

- delaware_NLCD_flat_042400.bin

Step 2: Open the FGDC File

We will copy information from the FGDC file into settings in the Manifold importer. To make this easier, we open the delaware_FGDC.txt file in Notepad or some other text editor. This file provides lots of metadata information for this data set.

Step 3: Launch Manifold and Import File

Without closing Notepad, we launch Manifold, open a project and choose File - Import - Surface. In the Import dialog we choose the Raw Binary Files importer in the Files of Type box and browse over to where we have saved the delaware_NLCD_flat_042400.bin file. We open it.
The Import Raw Binary File dialog opens. We will adjust some of the values in this dialog using values cited in the FGDC text file we have open in Notepad. Using the Windows task bar we will switch to Notepad, copy a value, switch to Manifold and paste the value into the Import Raw Binary File dialog.

Copy the value next to Row_Count and paste it into the Import Raw Binary File dialog's Height box.

Copy the value next to Column_Count and paste it into the Import Raw Binary File dialog's Width box.
Verify the number of Channels is set to 1 and Type is set to Integer (8-bit) and then press OK.

The result is a new surface in the project pane.

**Step 4: Open Surface and Specify Projection Information**

Open the surface by double-clicking on it. It is shown by default as an exquisitely detailed gray scale image. For best appearance, use View - Display Options to turn off Shading and Autocontrast.
Since the surface was not in a georeferenced format it has been imported into the default Orthographic projection. The first time the surface is opened, Manifold will raise an info bar as seen above to warn us that the projection has not yet been verified. In this case, we will use the same dialog used to verify the projection to manually add projection information.

To do so, we click on the info bar to launch the Edit - Assign Projection dialog. We can get the required projection information from the FGDC text file we have open in Notepad and enter it into the dialog.

Launch the **Assign Projection** dialog by clicking on the information bar shown in the surface window.
From the metadata we know to use the Albers Conical Equal Area projection, the ellipsoid (not shown in the screen shot but visible in Notepad), the Center Latitude, Center Longitude and other parameters for the projection dialog.
We can copy the projection parameters value from Notepad and paste into the Coordinates dialog as shown above. The Abscissa Resolution value goes into the Local scale X box and the Ordinate Resolution value goes into the Local scale Y box.

Press OK and the surface will be correctly imported and georegistered. Henceforth the surface and any images created from it will be georeferenced.

NCLD Classification Codes

The classification codes are imported into a surface as "heights" when we create a surface from raw binary data.

**Water**

11 Open Water

12 Perennial Ice / Snow

**Developed**

21 Low Intensity Residential

22 High Intensity Residential

23 Commercial / Industrial / Transportation

**Barren**
31 Bare Rock / Sand / Clay
32 Quarries / Strip Mines/Gravel Pits
33 Transitional

**Forested Upland**
41 Deciduous Forest
42 Evergreen Forest
43 Mixed Forest

**Shrubland**
51 Shrubland

**Non-Natural Woody**
61 Orchards/ Vineyards/Other

**Herbaceous Upland Natural / Semi-natural Vegetation**
71 Grasslands / Herbaceous

**Herbaceous Planted / Cultivated**
81 Pasture / Hay
82 Row Crops
83 Small Grains
84 Fallow
85 Urban / Recreational Grasses

**Wetlands**
91 Woody Wetlands
92 Emergent Herbaceous Wetlands
Import a VMAP Level 1 File

VMAP Level 1 is the most detailed public domain digital map of international regions currently available. See the Import Drawing - VMAP topic for a discussion of VMAP. VMAP levels 0 and 1 are products of the former National Imagery and Mapping Agency (NIMA) of the United States, renamed in a fit of bureaucratic irrelevance to the National Geospatial-Intelligence Agency, NGA. NIMA used to provide VMAP for free download. NGA may or may not make VMAP available for download.

Step 1: Download the Data

Visit the NGA site at http://www.nima.mil and drill down to their geodata downloads page (currently at http://geoengine.nima.mil). NIMA may choose to change their site at any time so this topic does not provide a detailed description of how to locate and download data from NIMA's site. If for any reason NGA no longer provides VMAP, users should file a Freedom of Information Act request to obtain this data.

Step 2: Unzip / Extract the Files

A .gz file is a "gnu-zipped" file that uses open source software to perform compression similar to Windows .zip files. .tar files are UNIX files that are created by the TAR ("tape archiver") program, which performs a function in UNIX similar to that of the arj program in DOS and UNIX. TAR can take many files organized in directories and combine them into a single file. It's a handy way of bundling related files together for transmission. .tar files must extracted into the many smaller files that comprise them.

When we download v1155.tar.gz from NIMA's site, what we've downloaded is a .tar file (a collection containing numerous files inside the .tar file) that has been further compressed into a .gz file. We can decompress the .gz and also extract the .tar file it contains using WinZip.

To do this, place the v1155.tar.gz file into a folder and double click it [assuming, of course, you've already installed WinZip on your system]. Note: before using WinZip to decompress files of this type, launch WinZip and make sure that WinZip's Option - Configuration - Miscellaneous tab does not have a check for "TAR file smart CR/LF conversion."

WinZip will open the .gz and see the compressed .tar file it contains. It asks us if we want to decompress it and open the .tar file. Click Yes. The following decompression process may take a while on a slow machine. WinZip will then display all of the files in the .tar file. Extract them all to a convenient directory, preferably named for the originating file. For example, we would extract all of the files from v1155.tar.gz into a directory called v1155.

Step 3: Import into Manifold
Launch Manifold. Before importing, create a folder in the project pane named v1155. Importing VMAP results in a huge number of files, so it is best to keep them organized within their own folder in the project pane.

To import, choose **File - Import - Drawing** and choose **VMAP Files (dht)** in the Files of Type box. Browse over to the folder where you placed the files extracted from v1155.tar.gz, browse down into the directory tree into the soamafr folder and double-click on the dht file that is located there.

The dht file is a catalog of what is available within the numerous files in various subdirectories below it. The **Import VMap Data Set** dialog allows us to choose which data we wish to import. For this example, we will choose Hydrography and Transportation and press OK.

A seemingly endless list of components is created in the project pane within two folders named Hydro and Trans. Two maps named Hydro and Trans are created as well. The two maps group the various hydro and trans drawings within a map for convenience. We can open these maps or open individual drawing or labels components.
The 1155 file covers a very large area because it includes a data set that is normally provided on the VMAP1 155 CD. The illustration above shows the hydro map zoomed in to central Panama with all layers but the Hydrotxt_lib_155_hydro layer turned on.

Applying some formatting can help comprehension. The scene above shows Costa Rica near the border with Panama. The lines in the WatcrsI_lib_155_hydro layer have been formatted in blue color.
VMAP1 imports into Manifold as georegistered drawing so it is easy to combine them with other georegistered components in maps. For example, we can download and import the DTED Level 0 data for this region and drag and drop the resultant surfaces into the map to create the scene above. We’ve also formatted the labels for greater visibility.
Query Templates

This topic provides templates that show the structure of generic queries that may be applied in individual situations.

Replace the meta names in angle < > brackets with the appropriate items to create real queries from these templates. Field names created by the query are given in **boldface** within square [] brackets. Otherwise, square brackets indicate optional items.

Note that SQL is case insensitive. The style below uses the traditional CAPITALS for SQL words, but this is not necessary. SQL is also insensitive to white space such as tabs, space characters and new lines so that one may adopt whatever writing style format is judged most legible and understandable. The style below uses traditional formatting.

Simple Query

```
SELECT <fields> FROM <tables> WHERE <condition> GROUP BY  <fields> HAVING <criteria> ORDER BY <expression>;
```

Show New Columns Created with Computations

```
SELECT <field1>+<field2> AS **Sum**, <field1>*<field2> AS **Product** FROM <table>;
```

Using Aggregate Functions

```
SELECT Max(<field>), Min(<field>), Avg(<field>) FROM <table>;
```

Total Records

```
SELECT Count(*) FROM <table>;
```

Total Records with Condition

```
SELECT Count(*) FROM <table> WHERE <condition>;
```

Linked Tables

```
SELECT DISTINCT <table1>.* FROM <table1> INNER JOIN <table2> ON <table1>.<field1> = <table2>.<field2>;
```

Linked Tables with Condition

```
SELECT DISTINCT <table1>.* FROM <table1> INNER JOIN <table2> ON <table1>.<field1> = <table2>.<field2> WHERE <condition>;
```

Grouping by Field

```
SELECT <field1>, Count(<field2>) FROM <table> GROUP BY <field1>;
```

Grouping with Filter

```
SELECT <field> FROM <table> GROUP BY <field> HAVING Count(*) > 1;
```
Text Pattern Matching

SELECT * FROM <table> WHERE <field> LIKE "A%";

Text Pattern Matching using Regular Expression

SELECT * FROM <table> WHERE <field> LIKEEX "[A-F].+";

Filtering by Certain Values

SELECT * FROM <table> WHERE <field>
IN (<value1>, <value2>, <value3>);

Filtering by Subquery

SELECT * FROM <table1> WHERE <field1> > ANY
(SELECT <field2> FROM <table2> WHERE <condition>);

Ordering by Field

SELECT * FROM <table> ORDER BY <field> ASC;

Top Records

SELECT TOP <number_of_records> FROM <table>
ORDER BY <field> DESC;

First and Last Value

SELECT First(<field>) AS [First], Last(<field>) AS [Last]
FROM <table>;

Parameterized Query

PARAMETERS <parameter> <type>;
SELECT <fields> FROM <tables> WHERE <condition-with-parameter>;

Update Query

UPDATE <table> SET <field>=<expression> WHERE <condition>;

Drop Table

DROP TABLE <table>;

Create Table

CREATE TABLE <name> (<field1> <type>, <field2> <type>);

Add Table Field

ALTER TABLE <name> ADD <field> <type>;
Add Table Field with Default Value

ALTER TABLE <name> ADD <field> <type> DEFAULT <default>;

Change Table Field Type

ALTER TABLE <name> ALTER <field> <type>;

Rename Table Field

ALTER TABLE <name> RENAME <field> TO <newName>;

Drop Table Field

ALTER TABLE <name> DROP <field>;

Delete Records from Table

DELETE FROM <table> WHERE <condition>;

Insert Records into Table

INSERT INTO <table> VALUES (<value1>, <value2>);

Select Records into New Table

SELECT <fields> INTO <newTable> FROM <table> WHERE <condition>;
Sample Queries

The *Nwind* sample database provided on CD with Manifold System is a version of the standard Microsoft *Northwind.mdb* database that is provided with Microsoft Access. It contains numerous tables and queries that may be used to learn database and query concepts.

This topic lists queries that are part of the *Nwind* database, with some minor adjustments for Manifold SQL.

**AutoLookup**

```
SELECT DISTINCTROW Orders.[Order ID], Orders.[Order Date], Orders.[Customer ID], Customers.[Company Name], Customers.Address, Customers.City, Customers.Region, Customers.[Postal Code], Customers.Country, Customers.Phone FROM Customers INNER JOIN Orders ON Customers.[Customer ID] = Orders.[Customer ID] ORDER BY Orders.[Order ID];
```

**Catalog**

```
SELECT DISTINCTROW [Categories].[Category Name], [Products].[Product Name], [Categories].[Description], [Products].[Product ID], [Products].[Quantity Per Unit], [Products].[Unit Price], [Products].[Discontinued] FROM [Categories] INNER JOIN [Products] ON [Categories].[Category ID] = [Products].[Category ID] WHERE ([Products].[Discontinued] = No) ORDER BY [Categories].[Category Name], [Products].[Product Name];
```

**Category List**

```
SELECT DISTINCTROW Categories.[Category Name], Categories.[Category ID] FROM Categories ORDER BY Categories.[Category Name];
```

**Category Sales for 1993**

```
SELECT DISTINCTROW [Sales for 1993].[Category Name], Sum([Sales for 1993].[Product Sales]) AS [Category Sales] FROM [Sales for 1993] GROUP BY [Sales for 1993].[Category Name];
```

**Customer List**

```
SELECT DISTINCTROW Customers.[Company Name], Customers.[Customer ID] FROM Customers ORDER BY Customers.[Company Name];
```

**Daily Order Totals**

```
SELECT DISTINCTROW Orders.[Order Date], Count(Orders.[Order ID]) AS [CountOFOrder ID], Sum([Order Subtotals].Subtotal) AS SumOfSubtotal FROM Orders INNER JOIN [Order Subtotals] ON Orders.[Order ID] = [Order Subtotals].[Order ID] GROUP BY Orders.[Order Date];
```

**Employee List**

```
SELECT DISTINCTROW Customers.[Company Name], Customers.[Customer ID], Customers.Region, Customers.[Postal Code], Customers.Country, Customers.Phone FROM Customers ORDER BY Customers.[Company Name];
```
Examples

SELECT DISTINCTROW [Last Name] & ", " & [First Name] AS [Employee Name], Employees.[Employee ID]
FROM Employees
ORDER BY [Last Name] & ", " & [First Name];

Employee Sales by Country (Parameter)

PARAMETERS [Beginning Date] DateTime, [Ending Date] DateTime;
SELECT Orders.[Order ID], [Last Name] & ", " & [First Name] AS Salesperson, Employees.Country, Orders.[Shipped Date], [Order Subtotals].Subtotal AS [Sale Amount]
FROM Employees INNER JOIN (Orders INNER JOIN [Order Subtotals] ON Orders.[Order ID] = [Order Subtotals].[Order ID]) ON Employees.[Employee ID] = Orders.[Employee ID]
WHERE ((Orders.[Shipped Date] Between [Beginning Date] And [Ending Date]))
ORDER BY [Last Name] & ", " & [First Name], Employees.Country, Orders.[Shipped Date];

Employee Sales for 1993

SELECT DISTINCTROW SUM([Order Subtotals].Subtotal) AS [Employee Sales], [First Name] & " " & [Last Name] AS [Employee Name], Employees.[Last Name], Employees.[First Name]
FROM Employees INNER JOIN ([Order Subtotals] INNER JOIN Orders ON [Order Subtotals].[Order ID] = Orders.[Order ID]) ON Employees.[Employee ID] = Orders.[Employee ID]
WHERE ((Orders.[Order Date] Between #01/1/93# And #12/31/93#))
GROUP BY [First Name] & " " & [Last Name], Employees.[Last Name], Employees.[First Name];

Freight Charges

SELECT DISTINCTROW Shippers.[Company Name], Orders.[Order ID], Orders.Freight, Orders.[Shipped Date]
FROM Shippers INNER JOIN Orders ON Shippers.[Shipper ID] = Orders.[Ship Via]
WHERE ((Orders.[Shipped Date] Between #03/25/93# And #03/31/93#))
ORDER BY Shippers.[Company Name], Orders.[Order ID];

Invoices

SELECT DISTINCTROW Orders.[Shipped Date], Orders.[Ship Name], Orders.[Ship Address], Orders.[Ship City], Orders.[Ship Region], Orders.[Ship Postal Code], Orders.[Ship Country], Orders.[Customer ID], Customers.[Company Name], Customers.[Address], Customers.City, Customers.Region, Customers.[Postal Code], Customers.Country, Employees.[Last Name], Employees.[First Name], [Order Details].[Order ID], Orders.[Order Date], Orders.[Required Date], Orders.[Ship Via], [Order Details].[Product ID], Products.[Product Name], [Order Details].Quantity, [Order Details].[Unit Price], [Order Details].Discount, Orders.Freight
FROM Employees INNER JOIN ((Customers INNER JOIN Orders ON Customers.[Customer ID] = Orders.[Customer ID]) INNER JOIN (Products INNER JOIN [Order Details] ON Products.[Product ID] = [Order Details].[Product ID]) ON Orders.[Order ID] = [Order Details].[Order ID]) ON Employees.[Employee ID] = Orders.[Employee ID]
ORDER BY Orders.[Shipped Date];

Order Details Extended

SELECT DISTINCTROW [Order Details].[Order ID], Products.[Product Name], [Order Details].[Product ID], [Order Details].[Unit Price], [Order Details].Discount, [Order Details].Quantity, [Order Details].[Required Date], [Order Details].[Shipped Date], [Order Details].[Ship Via], [Order Details].[Order ID], Orders.[Order Date], Orders.[Required Date], Orders.[Ship Via], [Order Details].Quantity, [Order Details].[Unit Price], [Order Details].Discount, Orders.Freight
Details.Quantity, [Order Details].Discount, CLng([Order Details].[Unit Price]*[Quantity]*(1-[Discount])*100)/100 AS [Extended Price]
FROM Products INNER JOIN [Order Details] ON Products.[Product ID] = [Order Details].[Product ID];

Order Information

SELECT DISTINCTROW Orders.[Order ID], Orders.[Customer ID], Customers.[Company Name], Customers.Address, Customers.City, Customers.Region, Customers.[Postal Code], Customers.Country, Orders.[Ship Name], Orders.[Ship Address], Orders.[Ship City], Orders.[Ship Region], Orders.[Ship Postal Code], Orders.[Ship Country], Orders.[Ship Via], Orders.[Employee ID], Orders.[Order Date], Orders.[Required Date], Orders.[Ship Date], Orders.Freight
FROM Customers INNER JOIN Orders ON Customers.[Customer ID] = Orders.[Customer ID]
ORDER BY Orders.[Order ID];

Order Review

SELECT DISTINCTROW Orders.[Order ID], Orders.[Order Date], Orders.[Required Date], Orders.[Shipped Date], Orders.[Customer ID], Customers.[Company Name], Employees.[Last Name], Shippers.[Company Name] AS [Ship Via], [Order Subtotals].Subtotal, Orders.Freight, [Subtotal]+[Freight] AS Total
FROM Employees INNER JOIN (Shippers INNER JOIN (Customers INNER JOIN ([Order Subtotals] INNER JOIN Orders ON [Order Subtotals].[Order ID] = Orders.[Order ID]) ON Customers.[Customer ID] = Orders.[Customer ID]) ON Shippers.[Shipper ID] = Orders.[Ship Via]) ON Employees.[Employee ID] = Orders.[Employee ID]
ORDER BY Orders.[Order ID] DESC;

Order Subtotals

SELECT DISTINCTROW [Order Details].[Order ID], Sum(CLng([Unit Price]*[Quantity]*(1-[Discount])*100)/100) AS Subtotal
FROM [Order Details]
GROUP BY [Order Details].[Order ID];

Product List

SELECT DISTINCTROW Categories.[Category Name], Products.[Product Name], Products.[Product ID], Products.[Units In Stock], Products.[Reorder Level], Products.[Units On Order], Products.[Quantity Per Unit], Products.[Unit Price], Categories.Description, Products.Discontinued
FROM Categories INNER JOIN Products ON Categories.[Category ID] = Products.[Category ID]
WHERE ((Products.Discontinued=False))
ORDER BY Categories.[Category Name], Products.[Product Name];

Product Names and IDs

SELECT DISTINCTROW Products.[Product Name], Products.[Product ID]
FROM Products
ORDER BY Products.[Product Name];

Products and Suppliers

SELECT DISTINCTROW Products.[Product Name], Products.[Product ID], Products.[English Name], Products.[Category ID], Products.[Supplier ID], Products.[Quantity Per Unit], Products.[Unit Price], Products.[Units In Stock], Products.[Units On Order], Products.[Reorder Level],
Examples

Products.Discontinued, Suppliers.[Company Name], Suppliers.[Contact Name], Suppliers.City, Suppliers.Country, Suppliers.Phone
FROM Suppliers INNER JOIN Products ON Suppliers.[Supplier ID] = Products.[Supplier ID]
ORDER BY Products.[Product Name];

Products on Order

SELECT DISTINCTROW Categories.[Category Name], Products.[Product Name], Products.[Units In Stock], Products.[Units On Order], Suppliers.[Company Name], Suppliers.Phone
FROM Categories INNER JOIN (Suppliers INNER JOIN Products ON Suppliers.[Supplier ID] = Products.[Supplier ID]) ON Categories.[Category ID] = Products.[Category ID];

Quarterly Orders by Product (Crosstab)

TRANSFORM Sum(CLng([Order Details].[Unit Price]*[Quantity]*
  (1-[Discount])*100)/100)
SELECT Products.[Product Name], Orders.[Customer ID]
FROM Orders INNER JOIN
  (Products INNER JOIN [Order Details] ON Products.[Product ID] = [Order Details].[Product ID]) ON Orders.[Order ID] = [Order Details].[Order ID]
WHERE ((Orders.[Order Date] Between #01/1/93# And #12/31/93#))
GROUP BY Products.[Product Name], Orders.[Customer ID], Orders.[Order Date]
ORDER BY Products.[Product Name]
PIVOT "Qtr " & DatePart("q",[Order Date]) In
  ("Qtr 1","Qtr 2","Qtr 3","Qtr 4");

Quarterly Orders by Product (Make Table)

SELECT DISTINCTROW [Quarterly Orders by Product (Crosstab)].[Product Name], [Quarterly Orders by Product (Crosstab)].[Customer ID], [Quarterly Orders by Product (Crosstab)].[Qtr 1], [Quarterly Orders by Product (Crosstab)].[Qtr 2], [Quarterly Orders by Product (Crosstab)].[Qtr 3], [Quarterly Orders by Product (Crosstab)].[Qtr 4] INTO [Quarterly Orders]
FROM [Quarterly Orders by Product (Crosstab)];

Sales by Category

SELECT DISTINCTROW Categories.[Category Name], Products.[Product Name],
  Sum(CLng([Order Details].[Unit Price]*[Quantity]*(1-[Discount])*100)/100) AS [Product Sales]
FROM Orders INNER JOIN ((Categories INNER JOIN Products ON Categories.[Category ID] = Products.[Category ID]) INNER JOIN [Order Details] ON Products.[Product ID] = [Order Details].[Product ID]) ON Orders.[Order ID] = [Order Details].[Order ID]
WHERE ((Orders.[Order Date] Between #01/1/93# And #12/31/93#))
GROUP BY Categories.[Category Name], Products.[Product Name]
ORDER BY Categories.[Category Name], Products.[Product Name];

Sales by Date (Parameter)

PARAMETERS [Beginning Date] DateTime, [Ending Date] DateTime;
SELECT DISTINCTROW Orders.[Order ID], Orders.[Shipped Date],
  Customers.[Company Name], [Order Subtotals].Subtotal AS [Sale Amount]
FROM Customers INNER JOIN ([Order Subtotals] INNER JOIN Orders ON [Order Subtotals].[Order ID] = Orders.[Order ID]) ON Customers.[Customer ID] = Orders.[Customer ID]
WHERE ((Orders.[Shipped Date] Between [Beginning Date] And [Ending Date]))
ORDER BY Orders.[Order ID];

Sales by Product

SELECT DISTINCTROW Products.[Product ID], Orders.[Order Date],
    Sum(CLng([Order Details].[Unit Price]*[Quantity]*(1-[Discount])*100)/100) AS [Product Amount]
FROM Orders INNER JOIN (Products INNER JOIN [Order Details] ON Products.[Product ID] = [Order Details].[Product ID]) ON Orders.[Order ID] = [Order Details].[Order ID]
GROUP BY Products.[Product ID], Orders.[Order Date];

Sales by Sale Amount

SELECT DISTINCTROW [Order Subtotals].Subtotal AS [Sale Amount], Orders.[Order ID], Customers.[Company Name], Orders.[Shipped Date]
FROM Customers INNER JOIN (Orders INNER JOIN [Order Subtotals] ON Orders.[Order ID] = [Order Subtotals].[Order ID]) ON Customers.[Customer ID] = Orders.[Customer ID]
WHERE ((Orders.[Shipped Date] Between #10/1/93# And #12/31/93#))
ORDER BY [Order Subtotals].Subtotal;

Sales for 1993

SELECT DISTINCTROW Categories.[Category Name], Products.[Product Name],
    Sum(CLng([Order Details].[Unit Price]*[Quantity]*(1-[Discount])*100)/100) AS [Product Sales]
FROM Orders INNER JOIN ((Categories INNER JOIN Products ON Categories.[Category ID] = Products.[Category ID]) INNER JOIN [Order Details] ON Products.[Product ID] = [Order Details].[Product ID]) ON Orders.[Order ID] = [Order Details].[Order ID]
WHERE ((Orders.[Order Date] Between #01/1/93# And #12/31/93#))
GROUP BY Categories.[Category Name], Products.[Product Name];

Sales Totals

SELECT DISTINCTROW [Employees].[First Name] & " " & [Last Name] AS Name,
    Orders.[Employee ID], Count(Orders.[Order ID]) AS [CountOfOrder ID],
    Sum([Order Subtotals].Subtotal) AS SumOfSubtotal
FROM Employees INNER JOIN ([Order Subtotals] INNER JOIN Orders ON [Order Subtotals].[Order ID] = Orders.[Order ID]) ON Employees.[Employee ID] = Orders.[Employee ID]
GROUP BY [Employees].[First Name] & " " & [Last Name], Orders.[Employee ID];

Subquery

SELECT DISTINCTROW Customers.[Contact Name], Customers.[Company Name],
    Customers.[Contact Title], Customers.Phone
FROM Customers
WHERE ((Customers.[Customer ID] In (SELECT DISTINCTROW Orders.[Customer ID] FROM Orders
WHERE Orders.[Order Date] BETWEEN #04/1/93# AND #07/1/93#))));

Summary of Sales

4040
Examples

```
SELECT DISTINCTROW Orders.[Shipped Date], Orders.[Order ID], [Order Subtotals].Subtotal AS [Order Amount]
FROM Orders INNER JOIN [Order Subtotals] ON Orders.[Order ID] = [Order Subtotals].[Order ID];
```

Summary of Sales (Parameter)

```
PARAMETERS [Beginning Date] DATETIME, [Ending Date] DATETIME;
SELECT DISTINCTROW [Orders].[Shipped Date], [Orders].[Order ID], [Order Subtotals].[Subtotal], Year([Shipped Date]) AS [Year]
FROM [Orders] INNER JOIN [Order Subtotals] ON [Orders].[Order ID] = [Order Subtotals].[Order ID]
WHERE ([Orders].[Shipped Date] BETWEEN [Beginning Date] AND [Ending Date]);
```

Supplier List

```
SELECT DISTINCTROW Suppliers.[Company Name], Suppliers.[Supplier ID] FROM Suppliers
ORDER BY Suppliers.[Company Name];
```

Ten Most Expensive Products

```
SELECT DISTINCTROW TOP 10 Products.[Product Name] AS [Ten Most Expensive Products], Products.[Unit Price]
FROM Products
ORDER BY Products.[Unit Price] DESC;
```

Union Query

```
SELECT [Company Name], [City]
FROM [Suppliers]
WHERE Country = "Brazil"
UNION SELECT [Company Name], [City]
FROM [Customers]
WHERE Country = "Brazil";
```
Using SQL to Select Map Objects

This topic provides a collection of queries that select map objects. When such queries are based on spatial relationships such as proximity they are often called spatial queries.

The normal use of queries is to generate a table display showing the results of the query. If the query includes the ID field in its results, query tables can automatically select the drawing objects corresponding to each row in the query table based upon the setting of the Automatically select query records option in the Tools - Options - Miscellaneous dialog. When this option is checked (the default setting) queries will automatically select objects in the drawing.

That is, with this option on querying a drawing will automatically deselect any selected objects and then select all objects returned by the query. This is equivalent to making a selection using Replace mode. When a query includes more than one drawing, all previously selected objects in all of the drawings will first be deselected and then objects from the query will be selected in the first drawing for which an ID column is found in the query.

If the Automatically select query records option is not checked then running a query will generate a tabular display of the query results, but the objects in that query will not automatically be selected. We can then use such a query-generated table to manually select objects in the drawing with interactive mouse commands much as we could use a drawing's table to select objects in the drawing.

The first three examples show the basic mechanism of selecting objects in drawings using the content of fields in their associated tables.

SELECT States.* FROM States WHERE IsArea(States.ID) AND (SELECT Max(Area(Counties.ID)) FROM Counties WHERE Contains(States.ID, Counties.ID)) > 100;

Selects states that have at least one large (>100 area units) county from the drawing called states.

SELECT DISTINCT FastFood.* FROM FastFood, Streets WHERE FastFood.Type = "McDonalds" AND Streets.Name = "Rue Delaunay" AND Distance(FastFood.ID, Streets.ID) < 100;

Selects all McDonalds restaurants near the street named Rue Delaunay in the drawing streets. Note the 'DISTINCT' just before the 'FastFood.*' This suppresses duplicate fastfood entries, a necessary step when querying restaurants in Paris given the famous French interest in McDonalds and other haute cuisine.

When queries are based on spatial relationships such as proximity they are often called spatial queries. The following examples show queries that select objects in maps based on spatial relationships.

SELECT Stations.Name, Distance(Stations.ID, 12, "mi") FROM Stations;

This creates a tabular view where the first column shows the name of the station and the second column shows the distance in miles from this station to the station whose ID is 12.

SELECT Stations.* FROM Stations WHERE Distance(Stations.ID, 12, "km") Between 50 And 80;

This selects all stations whose distance to station 12 is between 50 and 80 kilometers.

SELECT Stations.* FROM Stations WHERE (SELECT Min(Distance(Stations.ID, Clients.ID, "mi")) FROM Clients WHERE Clients.Importance = "High") < 10;

SELECT Stations.* FROM Stations WHERE
Examples

(SELECT Max(Distance(Stations.ID, Clients.ID, "mi")) FROM Clients
    WHERE Clients.Importance = "High") < 10;

The first query selects all stations within 10 miles of any important client. The second query selects all stations
within 10 miles of important clients.

SELECT TOP 5 PowerPlants.* FROM PowerPlants, States WHERE
    (States.Name = "California" Or States.Name = "Nevada") AND
    Contains(States.ID, PowerPlants.ID) ORDER BY PowerPlants.Power;

This selects the 5 largest power plants in California and Nevada.

SELECT States.Name FROM States WHERE EXISTS
    (SELECT * FROM Rivers WHERE Rivers.Name = "Mississippi" AND
    Touches(States.ID, Rivers.ID));

This selects all states on the banks of the Mississippi.

We may also use SQL as a utility language to perform functions that might otherwise require a script. For
example, suppose we would like to determine the nearest (non-self) neighbor of points in a drawing. We can loop
through all points in the drawing and use the following to select the ID of the nearest neighbor of the point whose
ID equals X:

SELECT TOP 1 [ID] FROM [Drawing] WHERE [ID] <> X
ORDER BY Distance([ID], X);

Alternatively, we can execute the following query to list the nearest neighbor of each object:

SELECT [ID], (SELECT TOP 1 [Copy].[ID] AS [Nearest Neighbor] FROM
    [Drawing] AS [Copy] WHERE [Copy].[ID] <> [Drawing].[ID] ORDER BY
    Distance([Copy].[ID], [Drawing].[ID])) FROM [Drawing];

See Also

Selecting Objects with Queries

Spatial Extensions
Geocoding Extensions
Rank Columns / Decision Support System

Rank Columns are used to construct a Decision Support System query and to report the results. A rank column reports percentages. The Manifold Decision Support System (DSS) provides a way to select records using flexible criteria with queries. DSS results are reported as percentage values from 0 to 100 percent within a rank column in a table. Creating a rank column creates a DSS query, the results of which are reported in the column.

In this example we will open the Order Details table and find all "luxury" goods. We shall define luxury goods to mean those goods that are both "expensive" and "rare". Expensive goods, in turn, are defined to be those goods for which the unit price is "high" and rare goods are those goods for which quantity is "low."

Step 1: Import and open the table

Import the Order Details table from the Nwind.mdb sample database.

<table>
<thead>
<tr>
<th>Order Details *</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unit Price</td>
</tr>
<tr>
<td>22</td>
</tr>
<tr>
<td>21.35</td>
</tr>
<tr>
<td>27</td>
</tr>
<tr>
<td>43.7</td>
</tr>
<tr>
<td>66</td>
</tr>
<tr>
<td>30.7</td>
</tr>
<tr>
<td>34</td>
</tr>
<tr>
<td>4.2</td>
</tr>
<tr>
<td>11.3</td>
</tr>
<tr>
<td>16.2</td>
</tr>
<tr>
<td>7</td>
</tr>
</tbody>
</table>

In this example we show only the Unit Price and Quantity fields for brevity.

Step 2: Add a rank column

Choose Table - Add - Rank Column to add a rank column.
This command launches the **Add Rank Column** dialog, which is used to add criteria and to create the query that will generate this rank column. Begin by changing the name to **Luxury**. This will be the name of the rank column as it will appear in the table.

**Step 3: Add criteria**

We will add two criteria.

1. Click on the **New** button to add the first criterion.

In the **Add Criterion** dialog, enter **Expensive** as the name of this criterion. Choose **Unit Price** for the column and **High** for the Type of criterion. The pane on the right shows the criterion curve that is automatically generated by DSS based on the data in the table. In essence, we've defined a criterion to find expensive goods by finding unit prices that are high. Press **OK**. (See the Add Rank Column Dialog topic for detailed instructions on the **Add Criterion** dialog.)
Back in the **Add Rank Column** dialog, the result is a new criterion that is saved in the hierarchical diagram under the name of the column (field) it uses, **Unit Price**. If we were to expand the **Unit Price** level we would see a criterion called **Expensive** underneath it.

Click on the **New** button once more to add another criterion.

For this criterion we will use the name **Rare**. We choose **Quantity** for the field (column) to use and **Low** for the type. The pane on the right shows the criterion curve generated by DSS for records that have low quantity values. This is a way of finding "rare" goods in that their quantities are low. Press **OK**.

Back in the **Add Rank Column** dialog we now have two fields in use with for criteria, the **Quantity** and the **Unit Price** fields.

**Step 5: Drag and drop criteria to create a rank query**

We create rank queries by dragging and dropping criteria from the upper pane in the **Add Rank Column** dialog to the lower pane. We begin by expanding the levels in the upper pane so we can see the criteria from which we can choose.
We drag and drop the Rare criterion from the upper pane to the lower pane.

The Rare criterion appears in the rank query pane under Criteria. So far, we've built a query that selects all records that are "rare", that is, for which quantity is "low". To make this a more interesting query we will drag and drop the Expensive criterion from the upper pane into the lower rank query pane.
The **Expensive** criterion now joins the **Rare** criterion in the rank query pane. By default, new criteria are added using Boolean **OR** to create a combined query. So far, our query will find all goods that are either rare or are expensive. We would like to amend this query so it finds those goods that are both rare and expensive.

To do so, we double click into the **Junction** field and change it to **And**.
The new rank query is now equivalent to "Select all records that are Rare and Expensive." So far, so good; however, we would like to find only those goods that are very expensive. We can "nudge" a criterion to accentuate it or de-emphasize it by using the **Hedge** column.

Double clicking into the **Hedge** column for the Expensive criterion we can choose **very** as the hedge.
The result is a query that finds all records that are Rare and very Expensive. [Note: The Not field allows us to prefix each criterion with a Not Boolean modifier. So, we could find records that are Rare and Not very Expensive if desired.]

Press OK to add the rank column using this query to the Order Details table.

**Step 6: Admire result**

A new column called **Luxury** appears in the Order Details table. It is a rank column and formatted by default to show percentages with two digits of accuracy after the decimal point. A gray background indicates it is a special column.

<table>
<thead>
<tr>
<th>Unit Price</th>
<th>Quantity</th>
<th>Luxury</th>
</tr>
</thead>
<tbody>
<tr>
<td>36.6</td>
<td>4</td>
<td>100.00 %</td>
</tr>
<tr>
<td>97</td>
<td>3</td>
<td>100.00 %</td>
</tr>
<tr>
<td>55</td>
<td>4</td>
<td>100.00 %</td>
</tr>
<tr>
<td>203.5</td>
<td>4</td>
<td>100.00 %</td>
</tr>
<tr>
<td>123.79</td>
<td>2</td>
<td>100.00 %</td>
</tr>
<tr>
<td>55</td>
<td>4</td>
<td>100.00 %</td>
</tr>
<tr>
<td>263.5</td>
<td>2</td>
<td>100.00 %</td>
</tr>
<tr>
<td>99</td>
<td>4</td>
<td>100.00 %</td>
</tr>
<tr>
<td>55</td>
<td>1</td>
<td>100.00 %</td>
</tr>
<tr>
<td>52.5</td>
<td>4</td>
<td>100.00 %</td>
</tr>
<tr>
<td>52.5</td>
<td>4</td>
<td>100.00 %</td>
</tr>
<tr>
<td>81</td>
<td>1</td>
<td>100.00 %</td>
</tr>
<tr>
<td>55</td>
<td>2</td>
<td>100.00 %</td>
</tr>
<tr>
<td>53</td>
<td>3</td>
<td>99.90 %</td>
</tr>
<tr>
<td>53</td>
<td>2</td>
<td>99.90 %</td>
</tr>
<tr>
<td>53</td>
<td>3</td>
<td>99.90 %</td>
</tr>
</tbody>
</table>

We can click on the column head to sort the tables by the values in the Luxury column. If we do so, we can see that those records given a 100% ranking all have very high prices and low quantities. Further down in the table are those records with not quite so very expensive prices or perhaps slightly greater quantities. At the very end of the table are those products that are cheap and abundant.
Shortest Path over Land

Finding the shortest path between two points is a typical task. Usually we find such a shortest path through a network made up of road lines. At times we may want to find the shortest path between two locations that satisfies other criteria in a region where there is no road network. For example, we might want to find the shortest path between two points that does not cross any water bodies, or we may want to find the shortest path between two points that never rises above a given terrain elevation, avoids certain obstacles or otherwise is constrained.

This example considers the general problem of finding the shortest distance between two points when travel is allowed only within a specified region. In the example we will find the shortest distance between two points where travel must occur on land within Mexico.

The general solution to this and other similar problems is to first build a network that exists only within the allowed region. We can then use the network to find a shortest path. Most of the procedure’s steps are creating the network. Once we have the network the rest is easy. To build a network we will create a regular grid of points. We will then remove all points that do not lie on land. We then build the network using the remaining points.

This is an unusually long example that will seem very intricate to a newcomer. For experienced hands it is very rapidly done. Beginners should have faith that mastery of elementary maneuvers in Manifold will allow them to speed through tasks like this example.

Step 1: Import drawing and prepare drawing

We begin by importing the mexico_eg.mif sample drawing. Open the drawing and choose Edit - Change Projection to open the projection dialog.

![Projection Dialog]

Project the drawing into Orthographic projection using the default parameters offered by pressing the Suggest button. Press OK.
The drawing will now appear in Orthographic projection. It is important to project the drawing so default measurements appear in meters.

Later in this example we will use spatial overlays to transfer field values for objects over land. To facilitate use of spatial overlays, we will add a field to the drawing's table. To add a field, open the drawing's table, right click onto any column head and choose Add - Column from the context menu.

In the Add Column dialog, choose the name Land for the new field and press OK to create the new field using the default type of Integer (32-bit). [We could, of course, create this as a Boolean field if desired since we will use this field to flag land areas. We used the integer type out of sheer laziness because it is the default.]

A new column called Land appears in the table. We now will set the value of Land to 1 for all land areas.

To do this, we press CTRL-A to select all the records in the table. We could have also chosen Edit - Select All from the menu.
We double click into any of the **Land** cells and enter the value 1 and press **Enter**.

This enters the value 1 into the **Land** column for all selected records. Next, we need to set the Transfer Rule for this column. To do so, we right click onto the **Land** column head and choose **Transfer Rules** from the context menu.

In the **Column Transfer Rules** dialog we choose **Copy** for the **1 to N** rule. The transfer rule tells Manifold how to allocate the value in this column when it is transferred to other objects. The **1 to N** rule is used when one of the objects (such as an area) transfers its value to many other objects (such as points within that area). The **Copy** rule tells Manifold to take the value in the object and to copy it into all of the receiving objects.

**Step 2: Create a grid of points**

Click back onto the drawing to move the focus there. We will now create a grid of points in the drawing. Begin by pressing View - Grid.

The **Grid** dialog opens with values estimated from the size of the open drawing window. We will change the **spacing** parameter to 100000 meters. This will create a grid of lines separated by horizontal and vertical intervals of 100 km. That's a fairly coarse grid but suitable for our example. Create the grid by pressing the **Create** button. This creates a grid of line objects in the drawing.

The choice of a grid interval is an important matter for the approximations that follow. The finer the grid, the finer will be the network we build from it. The finer the network, the greater the accuracy of the shortest path approximation. However, the finer the grid the greater the network analytic task as well as all geometric tasks involved in the construction of the network. We **strongly** urge the new user to begin with very coarse grids to master the technique. One can later progress to denser grids in gradual steps to see where performance...
becomes unacceptably slow. Experts who are sure of their technique may use dense grids that could take a weekend of processing to perform some tasks for a final result.

The Grid dialog is a dual-purpose dialog. It can be used to show a grid of lines created by the system that "floats" above the drawing but is not a part of it. In addition, it can be used to create line objects in the drawing in the shape of the grid. We've used the Grid dialog to create a grid of line objects. However, we do not want to show a grid overlay in addition to that. So, we uncheck the Show grid box and press OK to exit the Grid dialog.

In the drawing we now have a grid of lines spaced at 100 km vertical and horizontal intervals. We will create points at their intersections.

To do so, we first select all of the lines. Do this by pressing in the Select Lines select objects button and pressing out the Select Areas and Select Points buttons. We can then press CTRL-T or choose Edit - Select By Type from the menu to select all of the lines. [This is such a common task that it is good to become accustomed to pressing CTRL-T for faster editing.]

In the Transform toolbar, choose the [Selection] as the scope and Points as the transform operator. This will create points at the ends of all lines in the selection.
Examples

The result is the creation of many points that are separated from each other by the same vertical and horizontal interval, 100 km, used to create the lines.

While the points are selected, we can use the Remove Duplicates transform operator to eliminate any duplicated points. There will be duplicated points since four lines come together at each intersection and thus creating a point at the end of each line results in coincident points at the intersections.

We don't need the lines anymore so we can delete them. The easiest way to do this is to press CTRL-T to select by type again (the selection modes should still be set to select lines only) and then press Delete to blow away the lines.

The result is a grid of points that are separated by 100 kilometers.

**Step 3: Eliminate points over water using Spatial Overlays**

Our next task is to eliminate all points that are not located over land areas. We begin the task by selecting all points and saving them (using the Selections pane) to a saved selection called **Points**. To do so, open the Selections pane.
Select all of the points by pushing in **Select Points** select objects button only and pressing **CTRL-T** to select by type. Save the selection into the Selections pane and name it **Points**.

We also need to save the areas. To do so, push in the **Select Areas** select objects button only and press **CTRL-T** to select by type. Save the selection into the Selections pane and name it **Areas**.

The above two steps are not illustrated. Note the use of **CTRL-T** for speed. We could have also used the **Edit - Select By Type** menu choice if preferred.

In the Selections pane we now have two saved selections. One saved selection is called **Areas** and the other is called **Points**. We will use these saved selections with the Spatial Overlay dialog.

Launch the **Spatial Overlays** dialog by choosing **Drawing - Spatial Overlays**. In the dialog choose the saved selection called **Areas** as the **Source** and the saved selection called **Points** as the **Target**. Choose **Areas to contained points** as the **Method**. This tells Manifold to transfer the field values in accordance with the Transfer Rules from those objects in the source set to those objects in the target. We will transfer the field values from the areas to all points within the areas. Press **OK**.

In complex maps or those with many intricate areas, spatial overlays can take a long time to run. The geometric computation to determine if a point is inside or outside a given area can be very complex. Remember that areas in Manifold can be topologically branched objects where a single area object can have holes and islands. The computation to deal with arbitrarily complex topology can be very time consuming. This particular example will take approximately five to fifteen minutes to compute on desktop machines in the 500Mhz to 1 gigahertz range. Increasing the number of points or the number or complexity of the areas will increase the time required.

The spatial overlays operation will copy the field values from the areas to the points they contain. Every point located over a land area will receive the value 1 in its **Land** column. All points over water will stay with the default value of 0 in their **Land** column which was placed there when they were created from the grid of lines using the **Points** transform.

To create a grid of points that exists over land areas, all we need do now is select all points with a value of 0 and delete them. There are many ways of doing this, but we will use a somewhat indirect method to illustrate a few more techniques in Manifold.

Open the drawing’s table and click on the **Land** column to sort the table by that column. Scroll down to the first record with a value of 1 in its **Land** column. Ctrl-click on that record to select it. Scroll down to the bottom of the table to the last record with a value of 1 in the **Land** column and Shift-CTRL-click on that record to select all the records between the two clicks, thus selecting all objects with a value of 1 in the **Land** column.
In the drawing window, we can see that this has selected all of the land areas as well as the points over land. To eliminate the points over water, we can choose **Edit - Select Inverse** (which selects only the points over water) and press **Delete**.

From here on in it is more convenient to work in a map window. We create a map from the *Mexico_eg Drawing* and add a drawing layer to the map that we call **Nodes**. We will also add a drawing layer we call **Links**. We will move the points from the Mexico drawing to the **Nodes** drawing layer. First, we select all of the points (using **CTRL-T** or **Edit - Select By Type**) and then we cut the points using **CTRL-X** or **Edit - Cut**, click on the **Nodes** layer tab and then paste the points using **CTRL-V** or **Edit - Paste**.

When pasting the points into the new layer, Manifold will raise a dialog asking us how to paste the fields for the points. We don't really need any of the fields any more so it doesn't matter whether we paste the fields or not. The result will be a new drawing layer filled with points that are all above land areas in Mexico, as seen in the illustration above.

**Step 4: Build a network**

Our task now is to use the grid of points to build a network. We will use the Distance Network transform to wire up links between nearby points. To use this transform sensibly we have to figure out what the right distance is to specify that will create links between neighboring points.
The Tracker tool can help us. We will use the tool to measure distances.

Using the tracker tool, we can measure the distance from one point to its diagonal neighbor. It turns out this distance is approximately 141 kilometers (as the more mathematically inclined reader already knew, since the grid of points is 100 kilometers apart). The illustration above shows the use of the tracker together with the Snap to achieve precise measurement of distance between points.

That distance measurement means that if we want the **Distance Network** to build links to the immediate horizontal, vertical and diagonal neighboring points we should use a distance of 150,000 meters. Click on the **Links** layer in the map so that the new lines will be created there. We then load the transform toolbar as seen above and press **Apply**.
The result is that the lines created by the **Distance Network** appear in the **Links** layer in our map. [For the illustration above we have also formatted the **Nodes** layer so that points in the layer are bright green dots.]

![Map with links and nodes](image)

Some housekeeping: The **Distance Network** solver will create duplicates of some links. We eradicate these by running the Remove Duplicates transform in the **Links** layer.

One other item of housekeeping: At least one link has been created over water, across the Sea of Cortez between the Baja peninsula and mainland Mexico.

We can zoom into the map, select the unwanted link and press the **Delete** key to delete it.
For a small map and a small network as used in this example it is easiest to delete any links created over water by hand. For more complex networks, we can use spatial overlays once more to transfer the value 1 to all links that are contained by the land areas. We can then remove all links with a value of 0.

**Step 5: Find the shortest path**

All of the above steps, for all of their apparent intricacy served only to create the network we will now use to find shortest paths. The geometry of the network captures the essential information, where are paths that go over land and where are water regions to be avoided. All the rest is just a trivial matter of finding the shortest path through a network.

To find the shortest path between two points we click on the **Nodes** layer and select two points. In the example above, we've selected one point at the tip of the Baja peninsula and another point at near the eastern end of Mexico.
We then click on Links and select all of the lines in that layer (we do this by pressing **CTRL-A** or choosing **Edit - Select All** since there are only lines in this layer).

We then load the transform toolbar with the settings seen above. This instructs Manifold to use the current selection (two points and a network of lines) with the Select Shortest Path transform operator to select the links that comprise the shortest path between the two points.

Press **Apply** and the shortest path will be selected.

To see the shortest path better, we can press **CTRL-C** or **Edit - Copy** to copy the links involved and then paste them into a new drawing layer called **Path**. We can then format the lines in **Path** in a larger size so they stand out better.
Discussion

Clearly, the above method results in an approximation to the "perfect" shortest path. As the grid of points and resultant network get denser the approximation comes closer and closer to the shortest path that would be obtained if the path were not constrained to travel over a network.

Note also that the above procedure can be generalized to build networks within given regions. For example, suppose we had a surface that showed the terrain elevations in Mexico. Perhaps we are interested not just in overland paths, but only in those overland paths that stay below an elevation of 1000 meters. We could use the surface to build contour areas. We could then use spatial overlays to mark all points in the grid that fall within contour areas that are lower than 1000 meters in elevation and eliminate higher points. If we build a distance network using the remaining points we will have a network that exists only in those overland locations that are below 1000 meters elevation.

Suppose we were interested in travel between random points and not just those points on our grid? Perhaps, for example, we study Mayan civilization and we want to know the shortest paths between various Mayan cities. These are easy to add: after creating a regular grid of points we simply copy the points showing the locations of the cities and add them to the layer in which the grid of points is located. We then create a distance network using the combined grid of points and the irregularly arranged city points. To find shortest paths we can then use our layer of city points to select start and end points. Since copies of these points were part of the set of points used to create the distance network we know that each of the city points is correctly located at the end of links in the network.

This procedure describes a highly interactive approach to creating and using networks for special purposes. We might want to add some analytic dimension, such as a Viewbot that reports the sum of the lengths of lines in the selection. If we select the shortest path using the Select Shortest Path transform we can then see immediately the length of the shortest path.

If we like, we might copy the shortest path and place it in its own layer where we can run the Join Lines transform to create one line from the path.

We could then create a label using the Length (l) intrinsic field to label the shortest path with the length, as seen above. If we had many shortest paths in our map, such as the various shortest paths between cities, we could label all of the shortest paths with their lengths.

We used the above example to mark points that occurred over a relatively simple landform. The same procedure would have worked fine had the landforms included lakes or other more complex water features.
Extract Last Names using Tokens

This example uses Transform toolbar token operators to create a new column in a table that extracts the last name from another column that contains a full name. We will work with a table associated with a drawing that shows US congressional districts.

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Party</th>
<th>State</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>17343</td>
<td>Collin C. Peterson (D)</td>
<td>D</td>
<td>MN</td>
<td><a href="http://www.house.gov/">http://www.house.gov/</a></td>
</tr>
<tr>
<td>17347</td>
<td>Dennis R. Rehberg (R)</td>
<td>R</td>
<td>MT</td>
<td><a href="http://www.house.gov/">http://www.house.gov/</a></td>
</tr>
<tr>
<td>17348</td>
<td>C. J. &quot;Butch&quot; Otter (R)</td>
<td>R</td>
<td>ID</td>
<td><a href="http://www.house.gov/">http://www.house.gov/</a></td>
</tr>
<tr>
<td>17349</td>
<td>Earl Pomeroy (D)</td>
<td>D</td>
<td>ND</td>
<td><a href="http://www.house.gov/">http://www.house.gov/</a></td>
</tr>
<tr>
<td>17350</td>
<td>Doc Hastings (D)</td>
<td>D</td>
<td>WA</td>
<td><a href="http://www.house.gov/">http://www.house.gov/</a></td>
</tr>
<tr>
<td>17351</td>
<td>George Nethercutt (R)</td>
<td>R</td>
<td>WA</td>
<td><a href="http://www.house.gov/">http://www.house.gov/</a></td>
</tr>
<tr>
<td>17374</td>
<td>Rich Larsen (D)</td>
<td>D</td>
<td>WA</td>
<td><a href="http://www.house.gov/">http://www.house.gov/</a></td>
</tr>
<tr>
<td>17394</td>
<td>Norman D. Dicks (D)</td>
<td>D</td>
<td>WA</td>
<td><a href="http://www.house.gov/">http://www.house.gov/</a></td>
</tr>
<tr>
<td>17403</td>
<td>Jennifer Dunn (R)</td>
<td>R</td>
<td>WA</td>
<td><a href="http://www.house.gov/">http://www.house.gov/</a></td>
</tr>
<tr>
<td>17404</td>
<td>Jim McDermott (D)</td>
<td>D</td>
<td>WA</td>
<td><a href="http://www.house.gov/">http://www.house.gov/</a></td>
</tr>
<tr>
<td>17405</td>
<td>Jay R. Inslee (D)</td>
<td>D</td>
<td>WA</td>
<td><a href="http://www.house.gov/">http://www.house.gov/</a></td>
</tr>
<tr>
<td>17408</td>
<td>Adam Smith (D)</td>
<td>D</td>
<td>WA</td>
<td><a href="http://www.house.gov/">http://www.house.gov/</a></td>
</tr>
<tr>
<td>17421</td>
<td>Brian N. Baird (D)</td>
<td>D</td>
<td>WA</td>
<td><a href="http://www.house.gov/">http://www.house.gov/</a></td>
</tr>
<tr>
<td>17439</td>
<td>David R. Obey (D)</td>
<td>D</td>
<td>WI</td>
<td><a href="http://www.house.gov/">http://www.house.gov/</a></td>
</tr>
<tr>
<td>17448</td>
<td>James L. Oberstar (D)</td>
<td>D</td>
<td>MN</td>
<td><a href="http://www.house.gov/">http://www.house.gov/</a></td>
</tr>
<tr>
<td>17501</td>
<td>David Wu (D)</td>
<td>D</td>
<td>OR</td>
<td><a href="http://www.house.gov/">http://www.house.gov/</a></td>
</tr>
<tr>
<td>17507</td>
<td>Bart Stupak (D)</td>
<td>D</td>
<td>MI</td>
<td><a href="http://www.house.gov/">http://www.house.gov/</a></td>
</tr>
<tr>
<td>17534</td>
<td>Greg Walden (R)</td>
<td>R</td>
<td>OR</td>
<td><a href="http://www.house.gov/">http://www.house.gov/</a></td>
</tr>
<tr>
<td>17556</td>
<td>John Thune (R)</td>
<td>R</td>
<td>SD</td>
<td><a href="http://www.house.gov/">http://www.house.gov/</a></td>
</tr>
<tr>
<td>17561</td>
<td>Darlene Hooley (D)</td>
<td>D</td>
<td>OR</td>
<td><a href="http://www.house.gov/">http://www.house.gov/</a></td>
</tr>
<tr>
<td>17563</td>
<td>Mike Simpson (R)</td>
<td>R</td>
<td>ID</td>
<td><a href="http://www.house.gov/">http://www.house.gov/</a></td>
</tr>
<tr>
<td>17565</td>
<td>Earl Blumenauer (D)</td>
<td>D</td>
<td>OR</td>
<td><a href="http://www.house.gov/">http://www.house.gov/</a></td>
</tr>
<tr>
<td>17566</td>
<td>Ron Kind (D)</td>
<td>D</td>
<td>WI</td>
<td><a href="http://www.house.gov/">http://www.house.gov/</a></td>
</tr>
<tr>
<td>17590</td>
<td>Mark W. Kennedy (R)</td>
<td>R</td>
<td>MN</td>
<td><a href="http://www.house.gov/">http://www.house.gov/</a></td>
</tr>
</tbody>
</table>

The table’s Name field lists members of the 109th Congress in the US. We plan on using this drawing and table with the Manifold map server in a web page. To make it easy for browsers to find a member of Congress with a simple Find command, we would like to have a column that lists the last name only for each member. We feel it will be easier for browsers to find by entering "Peterson" than to have to try to remember the first name as well.

Our task is therefore to extract just the last name from the Name field and to put it into its own column, called Last Name. We can use tokens and the transform toolbar to accomplish this step.

Step 1: Get ready

We open the table in a table window. Next, we make ready with a few housekeeping items:

- **Separate tokens with:** 

We open the Tool Properties pane using SHIFT-ALT-T (or by clicking View - Panes - Tool Properties in the main menu). The default setting of the Separate tokens with box lists the space character (the blank location to the left of the  ), a tab, a carriage return and a newline character as the default token separator characters. Thus, every group of characters separated by these whitespace characters will be treated as a token. If the Separate tokens with box does not include a space character, add it.
It will be more convenient to work with the table if we hide all fields that are not necessary. We click on View - Columns and in the Columns dialog uncheck all the boxes except Name. This will hide all columns except Name.

That's better. We will now add two text fields to the table. Right click onto the Name column and choose Add - Column from the context menu.

In the Add Column dialog we create a new column called temp that is an ANSI variable length text column. Press OK. We will use this column as a temporary working space.
We also need to create the **Last Name** column that will hold our results. Right click onto the **Name** column again and choose **Add - Column** once more to create another column.

In the **Add Column** dialog we create one more column called **Last Name** that is also an ANSI variable length text column. Press **OK**.

We now have two new columns in our table.

**Step 2: Remove last token**

We observe that the names can be thought of consisting as a sequence of tokens. For example, "**Collin C. Peterson (D)**" consists of four tokens: "**Collin**, "**C.**, "**Peterson**, and "**(D)**". We would like to move the next-to-last token into the **Last Name** column. Unfortunately, within Manifold there is no "Copy next-to-last token" command. There is, however, a **Copy Last Token from** operator. If we get rid of the "(**D)**" and "(**R)**" last tokens we can then use the **Copy Last Token from** operator to extract the last names.

We begin by copying all of the values from the **Name** column into the **temp** column.

To do so, load up the transform toolbar as seen above and press **Apply**.
Like magic, Manifold copies all of the Name values into the temp column.

To delete the "(D)" and "(R)" tokens at the end of the values in the temp column we use the Delete last token operator in the transform toolbar as seen above. Press Apply.
We now have a column of names in temp where the final token is the last name we seek.

**Step 3: Copy last token into last name field**

To copy the last token we use the **Copy Last Token from** operator in the transform toolbar to copy the last token from the temp field into the Last Name field. Press **Apply**.

It seems we are done… but are we?

**Step 4: Correct mistake**

Unfortunately, we did not notice that the table contains names of the form "Henry E. Brown, Jr. (R)". For such records, copying the last token in the step above results in a Last Name field that contains the value "Jr.". No need to panic: we will find all such records and repair them.

To find all such records, we click on the Last Name column header to sort the column and then we scroll down to the section where all of the "Jr." records are located. We want our repair action to operate only selected records, so we will select all of the fields with "Jr." in the Last Name field. A fast way to do this is to click on the record handle of the first such record to select it.
We can then \textit{SHIFT-click} on the record handle of the last "Jr." record.

That will select all records from the previously selected record to the record where the \textit{SHIFT-click} occurred. We now have selected all the records with "Jr." in the \textit{Last Name} field. We could have used the Query toolbar to make this selection, but since we have all the records together in view in the table it was easier just to click and then shift-click.
The transform toolbar for tables uses **autoscope**: if any records are selected, it will automatically restrict the action of any transform operator to only the selected records. So long as only these records are selected, any operations we do in the transform toolbar will apply only to them.

We will begin by getting rid of the "Jr." tokens at the ends of the **temp** column values using the **Delete Last Token** operator as seen above. Press **Apply**.

This gets rid of the final "Jr." tokens at the ends of the fields in the selected records. Note that no changes were made to unselected records. However, we have a slight problem: there is a comma at the end of the selected names in the **temp** column. If we simply copy the last token into the **Last Name** field we will end up with some last names that have an unwanted comma at the end. We must first get rid of the commas.

To do this, we load up the toolbar with the **Delete Right** operator with a parameter of 1. This operator removes the last \(n\) characters on the right where \(n\) is the value given in the parameter box. Press **Apply**.
Nothing happens! The commas are still there.... what's going on? What's happened is that after we deleted the terminal "Jr." tokens a space character was left behind in the values in the temp field. When we used Delete Right with a parameter of 1 it deleted the space characters at the end of the strings. We need to apply Delete Right one more time.

As before, we load up the toolbar with the Delete Right operator with a parameter of 1. Press Apply.
Much better… we have now removed the final commas and are ready to copy the last names.

The final operation is to use **Copy Last Token from** to copy the last tokens from the **temp** field to the **Last Name** fields for the selected records.
The table is now almost in final form. All that remains is to right click onto the temp column head and choose Delete to delete the temporary field. We also deselect all records.

<table>
<thead>
<tr>
<th>Name</th>
<th>Last Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eddie Bernice Johnson</td>
<td>Johnson</td>
</tr>
<tr>
<td>Stephanie Tubbs Jones</td>
<td>Jones</td>
</tr>
<tr>
<td>Virgil H. Goode, Jr.</td>
<td>Goode</td>
</tr>
<tr>
<td>James A. Traicent, Jr.</td>
<td>Traicent</td>
</tr>
<tr>
<td>Robert L. Ehrlich, Jr.</td>
<td>Ehrlich</td>
</tr>
<tr>
<td>Harold E. Ford, Jr.</td>
<td>Ford</td>
</tr>
<tr>
<td>Robert E. &quot;Bud&quot; Cramer</td>
<td>Cramer</td>
</tr>
<tr>
<td>Frank Fallon, Jr.</td>
<td>Pallone</td>
</tr>
<tr>
<td>Henry E. Brown, Jr.</td>
<td>Brown</td>
</tr>
<tr>
<td>Sanford D. Bishop, Jr.</td>
<td>Bishop</td>
</tr>
<tr>
<td>F. James Sensenbrenner</td>
<td>Sensenbrenner</td>
</tr>
<tr>
<td>John Conyers, Jr.</td>
<td>Conyers</td>
</tr>
<tr>
<td>John M. Spratt, Jr.</td>
<td>Spratt</td>
</tr>
<tr>
<td>Walter B. Jones, Jr.</td>
<td>Jones</td>
</tr>
<tr>
<td>Jesse L. Jackson, Jr.</td>
<td>Jackson</td>
</tr>
<tr>
<td>John J. Duncan, Jr.</td>
<td>Duncan</td>
</tr>
<tr>
<td>Bill Pascell, Jr.</td>
<td>Pascrell</td>
</tr>
<tr>
<td>Felix J. Gucci, Jr.</td>
<td>Gucci</td>
</tr>
<tr>
<td>Ernest Jim Istook, Jr.</td>
<td>Istook</td>
</tr>
</tbody>
</table>

Our table now contains a Last Name field that has been extracted from the Name field.

Notes

This example used the default list of separators, where a space character denotes white space. A more sophisticated use of tokens might add other, non-whitespace characters to the token separator characters in the Tool Properties pane.

For example, suppose we have an address field containing text values such as “123 Main Street, Sioux Falls, South Dakota”. We could remove the space character from the token separator list and add the comma character “,” as a token separator. We could then copy the different parts of the address to different fields. In this case “123 Main Street” would be the first token, “Sioux Falls” would be the second and “South Dakota” would be the third token.
Import Landsat Files and Create Composite RGB Image

Before reading this example, review the Combining Channels into Images topic.

Landsat 7 images are often delivered as several .dat files that are accompanied by one or more header files. Each frequency band acquired by the satellite has one .dat file. For example, band 3 is Red, band 2 is Green and band 1 is Blue. The header files give information on the organization of the files.

In this example we import Landsat 7 Fast-L7A format .dat files to create a composite RGB image. We then alter the image to provide better contrast. Finally, we combine the RGB image with a higher resolution panchromatic image to provide an image that has both high resolution and color tinting. The result is an enhanced image that has much better readability than default R, G, B composites created using Landsat bands.

**Step 1: Determine image sizes**

By reviewing the header files (we can open them with Notepad or import them as comments components into Manifold) we can see the organization of the .dat files.

**PIXELS PER LINE = 1720 LINES PER BAND = 3322**

The header.hpr file in the line above tells us that most bands have 1720 width (pixels per line) and 3322 height (lines per band). The .hpr file defines parameters for bands 1, 2, 3, 4, 5, and 7.

Similarly, the header.hpn file tells us that the panchromatic band has 3440 height and 6644 width. This makes sense, since we know from the PIXEL SIZE parameter listed in the header files that most Landsat 7 bands provide 25 meter resolution while the panchromatic band provides 12.5 meter resolution. For the same area the panchromatic band therefore should have twice as many pixels as the other bands. The header.htm file provides information for the relatively low resolution bands 6h and 6l, which have pixels that are 50 meters in size. These bands are not used in this example.

**Note:** The example uses a recent Landsat 7 data set using the Fast-L7A data format. Older Landsat data has lower resolution, with 30 meter resolution in most bands and 15 meter resolution in the panchromatic band. Read the header files to see the exact specifications for your data set. The header.htm file provides information for the relatively low resolution bands 6h and 6l, which have pixels that are 50 meters in size.

**Step 2: Import .dat files**

Use File - Import - Surface with Files of Type set to Raw Binary Files (*.bin, *.dat, *) to import band1.dat.

In the Import Raw Binary File dialog we use the settings shown above with a Width of 1720 and a Height of 3322. Press OK. The result of the import is a new surface. Import band2.dat and band3.dat using the same settings.
When we finish the imports we will see three surfaces in our project file. The **Header** comment component was created when we imported the **header.hpr** file (not shown explicitly in this example) as a comment. We did this to find out the values of **Width** and **Height** to use for importing.

**Step 3: Create mask images**

We will now create grayscale mask images from the surfaces. These can then be combined in the R, G and B channels of an RGB image to create a composite color image.

Open the **Band1** surface. It appears as a speckled pattern. Choose **View - Display Options** to open the **Display Options** dialog.

In the **Display Options** dialog, uncheck the **Shading** and **Autocontrast** boxes. Press **OK**.

The result is a visible "image" seen in the surface window. It is a grayscale representation of the contents of **Band1**, the blue band from Landsat's Enhanced Thematic Mapper remote sensing instrument.

To save **Band1** as an image, we click on **Edit - Save Mask / Channel**.

![Image of the Display Options dialog with Shading and Autocontrast unselected](image-url)
In the **Save Mask / Channel** dialog we choose Channel: grayscale in the Save To box. In the To box we provide the name of the image to be created. We will enter Band 1 (Blue) so that later we will not be confused what the origin is of this image or what the intended usage of it is. Press OK and a new image called Band 1 (Blue) will be created. This new image is exactly the same size as the surface from which it was created: 1720 by 3322 pixels.

We can repeat the above process for the Band2 and Band3 surfaces. We open each surface in turn and use **Save Mask / Channel** to create an image from it.

The result of our work so far is that we now have three surfaces and three images created from them. We don't need the surfaces any more so if we are working on a machine with limited RAM we can delete them if desired.

**Step 4: Create an RGB image and load R, G and B channels**

We will now create an RGB image and then populate the R, G and B channels of the image from the mask images we created in the previous step.

We begin by creating a new RGB image. Double-click open one of the mask images, such as Band 1 (Blue).

Next, choose File - Create - Image and enter the above values into the **Create Image** dialog. Note that the size of the image is specified as exactly the same size as the surfaces originally imported and the mask images created from them: 1720 by 3322 pixels.

**A digression:** We clicked open the Band 1 (Blue) image because any newly-created image will use whatever coordinate system is in use by an open image window that has the focus when the new image is created. This is a nuance that has no effect in this example as written because all images so far are in the default Orthographic projection; however, if we had georegistered the band surfaces when they were first imported (to, say, UTM projection normally used with Landsat images) we would want all subsequent components to be created using the same projection.
We open the new image and see it is entirely black (as a newly created image will be by default). We can now populate it with R, G and B channel information from the saved mask images.

Choose Edit - Load Mask/Channel and in the Load Mask/Channel dialog specify the red channel in the Load box and the Band 3 (Red) image in the From box. We use the default mode, Replace, so the contents of the mask override whatever is in the image's red channel. Press OK.
The red channel will now be populated in our image. Since only red values appear, the image will seem to be a reddish rendition of the Band 3 (Red) grayscale image.

We choose Edit - Load Mask/Channel once again and in the Load Mask/Channel dialog we specify the green channel in the Load box and the Band 2 (Green) image in the From box. Press OK.
The green channel will now be populated as well as the red channel. Since greens and reds together make yellows, the image appears mostly in yellow tones.

Yet again we choose **Edit - Load Mask/Channel** and in the **Load Mask/Channel** dialog we specify the **blue** channel in the **Load** box and the **Band 1 (Blue)** image in the **From** box. Press **OK**.
With all three channels populated the image appears in "natural" shades. Like many Landsat images it seems to have poor contrast as if seen through a hazy atmosphere. Image processing within Manifold and use of other bands can improve the contrast.

**Step 5: Process image for better appearance**

If desired, we can open each of the band images, such as **Band 1 (Blue)**, and adjust their brightness and contrast. We can then load the R, B and G channels again using the adjusted band masks.

Opening the **Band 1 (Blue)** image we see it does not have a very great contrast variation.
Using the Brightness / Contrast command we can increase the contrast and decrease the brightness slightly. We will also open the Band 2 (Green) image and increase the contrast on that image as well.

We can also import band7.dat which provides an image using infrared frequencies. We will import band7.dat and create a mask image from it as we did for the other three bands in step 3 above. We can then open the resultant mask image and increase the contrast on that image.

Next, we repeat step 4 above to re-load the R, G and B channels of the composite RGB image. We use the increased contrast blue and green mask images and the new, increased contrast infrared image for the red channel. This achieves a composite image with enhanced appearance.
If desired, we can adjust the Brightness / Contrast on this image and also increase the saturation using Hue / Saturation to achieve even greater definition.

**Step 6: Incorporate panchromatic band for increased resolution**

We can increase the resolution of the image yet further by combining information from the panchromatic Landsat band. The Landsat bands used so far have a resolution of 25 meters per pixel.

<table>
<thead>
<tr>
<th>Local offset</th>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Local scale</td>
<td>25.00</td>
<td>25.00</td>
</tr>
</tbody>
</table>

We can specify this for the composite image by opening the Edit - Assign Projection dialog and specifying 25 meters as the X and Y Local scale.

However, the panchromatic instrument in Landsat creates a grayscale image using green to infrared wavelengths that has 12.5 meter resolution. We can import this surface from band8.dat using a setting of 3440 for width and 6644 for height. Like the other images, we can create an image from the surface using Edit - Save Mask/Channel. We can then open the mask image and increase the contrast slightly.
The result is seen above, zoomed into the same area shown in the other illustrations. Clearly, the panchromatic instrument provides better resolution. We can exploit this enhanced resolution by using the panchromatic image together in a map with the composite RGB image.

To do so, we first have to tell Manifold the pixel size of the panchromatic image. We do this by opening Edit - Assign Projection and specifying 12.5 meters as the X and Y Local scale.

Next, we create a map component using the panchromatic image, we open the map and then we drag and drop the composite RGB image into the map. Drag the composite image layer to below the panchromatic layer. Change the opacity of the panchromatic image to 70% so the colors from the composite image underneath can show through. Finally, open the composite image in its own window and increase the saturation using Hue / Saturation to a very high value so that the colors seen in the map window will be reasonably clear even though they are seen through the panchromatic grayscale image.
The result (seen above) provides visual cues from the high resolution grayscale panchromatic image that are tinted by the colors from the composite RGB image seen from below. Our eye is assisted by the black and white detail from the high-resolution image into seeing a clearer picture of the landforms.

However, the image could have better contrast. There are several ways to accomplish increased contrast. One way is to alter the contrast of the panchromatic image and to adjust the opacity to allow more or less of the RGB image to show through. A different way is to merge the panchromatic image and the RGB image into a single image and to then make contrast adjustments to that combined image.

To merge our map view of the partially transparent panchromatic image and the RGB image into a single image, with the map view open we create a layout for the map, and then the Tools - Make Image command to create a single image from the map at whatever resolution is desired. Note that Make Image preserves the coordinate system (projection) of whatever component it images.

We can then open the resulting image and apply commands such as Autocontrast in the transform toolbar to create the result seen above. Clearly, it's a long way from the original assembly of R, G and B channels as seen below.
**Step 7: Georegistering the image**

When Landsat bands are imported as surfaces they are imported using the default Orthographic projection with a scale of one pixel per meter. All components created from these original surfaces will have the same projection. To superimpose different bands having different pixel sizes upon each other in a map we must at least inform Manifold of the scales for each image using the **Edit - Assign Projection** dialog as done above.

To use the images in any geographic context we must also tell Manifold what projection is to be used. If we know the projection (from, for example, a reading of the header files) we can specify the projection in **Edit - Assign Projection** as well. The image will then be georegistered.

If the Landsat image is to be correctly georegistered, this would actually be one of the first steps in the process, immediately upon importing the .dat files. However, since many users will employ a Landsat image purely for visual effect and don't care whether it is registered or not we have placed this step at the end of the sequence.

To georegister the Landsat image we need to know what projection was used for it. From the header.hrf file giving specifications for all bands except 8 and 6h and 6l we read:

GEOMETRIC DATA MAP PROJECTION =UTM
ELLIPSOID =GRS80
DATUM =GDA
USGS MAP ZONE = 55
LL = 1462457.5412E 432422.2444S 452712.500 5193912.500

The LL values provide the coordinate location of the lower left corner of the display with the first pair of numbers given in degrees - minutes - seconds and the second being the offset in meters within the UTM zone 55 projection coordinate system. This magic information may be found through careful reading of documentation for Landsat.

To georegister this image, all we need do is correct the projection information loaded by default (Orthographic) into what the projection of the image actually is. To do this, we use the **Edit - Assign Projection** dialog.

In this projection we enter

**Universal Transverse Mercator - Zone 55 (S)** as the projection. The datum is Geodetic Reference 1980. The Local offset values are 452712.50 and 5193912.50. The Local scale values are both 1.00. The Scale correction values are 0.9996 and the False easting/northing is 500000.00 for X and 10000000.00 for Y.

Where did these values come from? Some came from the selection of the projection zone and some we added as follows:

From the header.hrf file we know the projection is UTM zone 55. We know that this is the UTM zone from the Universal Transverse Mercator (South) collection of UTM projections because the LL value gives a South latitude in the degrees - minutes - seconds pair of numbers for LL. Expert readers will immediately realize the Landsat image in question is in the Southern Hemisphere (it shows a forest region in Tasmania).

The datum is set to Geodetic Reference 1980 using the ellipsoid and datum information in header.hrf as well. Sometimes when parsing government header files it is important to know what the various abbreviations (such as GRS80 and GDA) mean. Enough digging into documentation available on Internet will usually provide answers.

The Local offset values are taken directly from the second pair of LL values. The values for Local scale are taken from the PIXEL SIZE value given in header.hrf. This pixel size will be different for the other bands.

**Notes**

The above commentary on georegistering a Landsat image has been placed in Step 7 at the end of this topic in order not to confuse those readers who will be working with Landsat images for purely visual presentation where georegistration is not necessary.

When users desire a georegistered Landsat image, most will georegister the bands right after they import them in Step 2. All subsequent components created from the bands (such as images used as masks or composite
images) will then also be georegistered. To assure the composite RGB image is georegistered, make sure one of the mask images is opened and has the focus when the new RGB image is created. The new, blank image will then be created using the same coordinate system as the image that currently has the focus.
Cookie Cutter a Large Image with Transfer Selection

This example shows how to cut a region in the shape of a particular country out of a larger image. The image used is one of NASA's "Blue Marble" images of the Earth as seen from space in natural color. We will create a smaller image that shows only the region within Afghanistan. This same method can be used to cut out images that fit other countries, such as the United States, or regions such as Europe. Like any operations with larger images the operations in this topic should be performed on a reasonably fast computer that is equipped with plenty of RAM.

NASA's Blue Marble series of images provides spectacular views of the Earth as seen in natural color. Visit http://visibleearth.nasa.gov and search for "Blue Marble" to find them. Since web page URLs frequently, using a good search engine such as Google with search terms like "NASA", "Blue Marble", "land_ocean_ice_8192.tif" and similar may be required to find the images.

This image and others from the Blue Marble series is on the Manifold download site as well.

**Step 1: Import and georegister the image**

This example uses a medium resolution image downloaded in TIF format from the NASA site. It is 8192 pixels wide and 4096 pixels high and named `Land_ocean_ice_8192.tif`. Import the image using File - Import - Image. For brevity, henceforth we will refer to this image as "the image" instead of using the long name.

According to the NASA web site the image is in Latitude / Longitude projection, that is, no projection.

We import the image into Manifold. This particular image is very useful for maps showing land regions since it already uses transparent pixels for all ocean regions.

Because the image has been imported from ordinary .tif format (which does not save projection information) and not from GeoTIFF format (which uses the same .tif extension but does save projection information), there is no projection information available to Manifold and so it is imported using Orthographic projection. Manifold recognizes that as a sign the projection may need to be manually assigned and so when we first open the image Manifold will display an info bar advising us the projection in use has not yet been verified.

Our first tasks are to verify the projection and to georegister the image, which in this case consists of correctly assigning the projection to be used. We can accomplish both tasks by clicking on the info bar and then entering the correct projection parameters into the Assign Projection dialog that pops up.

If the image is 8192 pixels wide and the width of the image is 360 degrees (which it must be to cover the entire Earth in Latitude / Longitude projection) we know that the local scale in the X (width, East - West) direction is:
Examples

\[
360 / 8192 = 0.0439453125
\]

If the image is 4096 pixels high and the height of the image is 180 degrees we know likewise that the local scale in the Y (height, North - South) direction is:

\[
180 / 4096 = 0.0439453125
\]

To georegister the image, we click on the info bar and then use these values in the Edit - Assign Projection dialog.

An experienced user will know that images from geographically unaware formats will be imported with positive coordinates only. However, an image in Latitude / Longitude projection will span both negative numbers and positive numbers in the range -180 to +180 degrees longitude and -90 to +90 degrees latitude. The coordinate system for the image must therefore be offset by -180 in \( X \) and -90 in \( Y \) so that half of the image uses negative numbers in longitude and latitude as occur in Latitude / Longitude projection.

An inexperienced user will not normally remember (or know how) to specify the Local offset numbers. Instead, the user will discover the above numbers after showing an image in a map together with an accurately georegistered drawing or other component. If Local offset is left at 0 for \( X \) and \( Y \) the image will be placed so that...
the lower left corner of the image is at the intersection of the Equator and the Prime Meridian. By changing the X local offset to -180 and the Y local offset to -90 the image is moved correctly into place. Inexperienced users usually find the correct Local offset numbers in a trial and error process like that shown in the Manually Georegister an Image topic.

**Step 2: Show in a map with a drawing**

We will import the Countries drawing from the Manifold CD. This is a drawing in Latitude / Longitude projection in the Countries lowres.map file. We import it using File - Import - Component to import the drawing from the .map file.

Create a map using the Land_ocean_ice_8192 image and open the map so created.

Next, drag and drop the Countries drawing into the map.

The Countries drawing appears in the map using default formatting.
Because the map was created using the **Land_ocean_ice_8192** image it is in **Latitude / Longitude** projection and the **Countries** drawing is also in **Latitude / Longitude** projection. Just to be sure the drawing is in exactly the same projection used by the map and by the image, we will right click onto the **Countries** tab in the map and choose **Project to Map**. This guarantees the drawing is in the same projection as the map and will assure maximum speed.

The image will be hidden by the areas in the drawing. We are interested in seeing only the boundaries of the areas so that the regions underneath the areas in the drawing can be seen.

To make the image visible, click on the background color well for areas and change the background color to transparent color.

The result is an upper drawing layer that has transparent color for area backgrounds. The effect is as if we replaced the area objects in the drawing layer with boundary lines. The image is now visible and can be seen to be perfectly georegistered.

If we would like to have the area border lines stand out more, we can change the area border line foreground color from the default dark gray to black.

Now is a good time to save a copy of the .map project.

**Step 3: Select image pixels in Afghanistan**

We now will select pixels in the image that are within Afghanistan. To do this we will first select the area representing Afghanistan in the drawing and then transfer the selection to the image.
Zoom in to Afghanistan. Because we changed the area background color to transparent color we don't see the "inside" of the areas. All we see are the boundary lines for the areas, but each country shown in the country drawing is represented by an area. We can click within the boundary lines of any country to select the area representing that country.

Using **Select Touch**, click on the Afghanistan area to select it. The entire central part of the Afghanistan area is seen in red selection color. The entire area is shown in red selection color because the drawing contains areas and not just boundary lines. It only appears to consist of boundary lines because we have changed the background, color of the areas to transparent color.

We will now start editing the image based on the selection in the drawing. Click on the image's layer tab in the map to make it the active layer.
To transfer the selection to the image we launch the Image - Transfer Selection command. We choose to modify the image and check the box for the Countries drawing in the Using pane. Press OK. There is no immediate change in the appearance of the map because the red selection color that was shown in the drawing is now shown in the same location for the region of selected pixels in the image.

We double-click on the Countries layer tab in the map to turn it off and then click on the image layer tab to make the image layer the active layer again. Now that the drawing is no longer displayed in the map we can see that the selection is a region of pixels in the image.

**Step 4: Crop image to selected pixels only**

Our objective is to create an image of Afghanistan only. To do that we begin by cropping the image to Afghanistan only and then we will delete all pixels except the selected pixels.

We will begin by cropping the image to just that rectangular extent that contains the selected pixels. There is no point in working with an overly-large image if all we are interested in are the pixels that comprise Afghanistan.
In the Transform toolbar, use the Crop operator on the selected pixels and press Apply.

We can see we have cut out the rectangular region of pixels containing Afghanistan from the much larger original image. The image is now only 328 pixels by 208 pixels and so requires far less size to store. The smaller image will also allow faster system performance.

We now delete all pixels outside of Afghanistan. Choose Edit - Select Inverse to select all pixels outside of Afghanistan.
Choose Edit - Delete to delete these pixels. The background appears white because we are viewing a map that has a default background of white (see the Layers pane topic). If we switched off the background we could see that the image consists of the pixels within Afghanistan and invisible pixels.

In images, "deleting" a pixel turns it into an invisible pixel. Note from the Invisible Pixels topic that invisible pixels are still there even though they cannot be seen.

If we turn on the Countries layer once more we can see that the only visible pixels in the image are those in Afghanistan. In the illustration above we have renamed the image Afghanistan.

Now would be a good time to save this .map file so that in the future we could import the image whenever we need a georegistered image of Afghanistan. We could use that image by itself to create cool new maps or we could use it with other images and drawings for special effects.
For example, the above map was created by showing the cropped Afghanistan image in a map above the Countries drawing and a lower layer using the original image. The lower image layer is shown with 40% opacity to make it less vivid than the Afghanistan image. In addition, a drop shadow created using Gaussian Blur was added below the Afghanistan image.

**Step 5: Create a drop shadow (for advanced users)**

Drop shadows are frequently used to provide a "floating" effect as though some parts of a map are floating above other parts. The following description is not central to the main theme of this Example (the theme is using Transfer Selection to crop images to match desired areas in drawings) but for advanced users it shows how to create a drop shadow. As this section is intended for advanced users, the descriptions assume greater familiarity with Manifold System.

Copy the Afghanistan image and paste it as a new image. Rename the new image Shadow. Open the image in an image window.
Use Add Margin in the transform toolbar to add 50 pixels of margin. We will need the extra margin to make a drop shadow since the image was cropped right to the edge of Afghanistan.

Using Touch Select, click on the black margin to select it. Press Delete to "delete" it (that is, turn it into invisible pixels).
We now have the same image but with a larger margin of invisible pixels around Afghanistan. We will color the invisible pixels white and the Afghanistan pixels black.

Use Image - Adjust - Hue / Saturation and move the **Lightness** slider all the way to zero to make the Afghanistan pixels black. The invisible pixels are unaffected by the **Lightness** slider.
In the format toolbar make sure the foreground color is white and then use **SHIFT** - Paint Bucket to click anywhere in the region of invisible pixels to color them white. The **SHIFT** key will cause the white color to fill all invisible pixels anywhere in the image.

Use **Gaussian Blur** to blur the image by 10 pixels. So far, so good, except that we don't want a white background, we want a transparent background. This is accomplished using a mask to change the alpha channel transparency in the **Shadow** image.

Copy the **Shadow** image and paste it as a new image. Rename that image **Mask**.
Open the **Mask** image and use Image - Convert To to change the mask image into a **Grayscale** image. Only grayscale images can be used as masks. Close the **Mask** image and click on the **Shadow** image again to make it the active window.

Use the Edit - Load Mask/Channel command to load the **Mask** image into the alpha channel of the **Shadow** image.
The final result is a Shadow image that is a Gaussian drop shadow of the original Afghanistan pixel region. The amazing thing about Manifold as opposed to ordinary graphics editors is that this image is still georegistered to the original Afghanistan image. After all, it was pasted from the original, georegistered image and so was georegistered when it was pasted. Manifold maintained its georegistration throughout the entire sequence of image editing commands that were applied.

To use this drop shadow, we drag and drop it into our map underneath the Afghanistan image. The shadow appears immediately underneath the Afghanistan image, but to give a convincing optical effect to fool the eye we should offset the shadow slightly lower and to the right.

To do this we right click on the Shadow layer tab and choose Register. In the Register dialog we can change the Local offset in X and Y to displace the drop shadow downward and to the right so that it is partially visible underneath the Afghanistan layer. We increased the X value slightly and decreased the Y value slightly, the exact changes being found by trying different values with the Preview box checked.

As seen together with the Countries drawing layer and the Afghanistan image layer the Shadow layer gives a convincing impression of a drop shadow. Note that because the Shadow layer employs alpha channel transparency, it will correctly allow any other layer beneath it to show through in a convincing manner.

For example we can add a copy of the original image layer underneath all other layers and set the opacity of the background image layer to 40% to de-emphasize it. (We can get this copy of the original image layer by importing it from the .map project saved earlier in this example.) We can also add a labels component. Finally, we can move the Countries layer below the Afghanistan image so that the image would not be ringed with a black boundary line, and we can lower the opacity of the Countries layer so that the black border line is also de-emphasized.
The completed map is seen above.

Comments

Drop shadows created in the above manner are remarkable things in that they show correct transparency as regards any layers below them. If other drawing layers (such as railroads or hydrography) were turned on underneath the Afghanistan and Shadow layers, the drop shadow would correctly drape over any drawing objects passing underneath the shadow just as shadows do in real life.

How is it that we could use the Shadow image as a mask to specify transparency? This is a fortunate coincidence in that the transparency of an alpha channel is set by the range from black (no transparency) to white (full transparency) that exactly coincides with the grayscale range of the Shadow image. Using the Shadow image in grayscale form as its own alpha channel mask gives us the transparency we want in the drop shadow.

The detail-oriented reader will note that creating a margin of 50 extra pixels results in a slightly larger image than necessary for the Shadow image. After all, we need only those pixels that extend to the edge of the darker pixels created by the Gaussian Blur yet using the procedure above we are creating a Shadow image that is slightly larger than necessary because it involves invisible pixels beyond the reach of the shadow effect.

We could have arranged for a smaller image by using the Crop command on the Shadow image. After doing the blur, we could have selected a rectangular section of pixels in the Shadow image that extended just past the darker pixels created by the Gaussian Blur command and then cropped the Shadow image to that rectangular section. This is not such a big deal with such a small image, but it is a fine point of detail that might bear attention in the case of much larger images.
Download and Mosaic TerraServer Images

At the present writing, Microsoft's terraserver-usa.com web site provides free downloads of USGS images in one meter resolution for the United States. TerraServer may be accessed either via an interactive web site visited by users, or as a web service to which applications like Manifold may connect to automatically fetch desired images. See http://terraservice.net or http://www.terraserver-usa.com for details. As always, URLs are subject to change and often will change after documentation like this is written. If the URLs in this topic are no longer valid, use a good search engine to hunt down the new locations.

Important: The very cool Microsoft TerraServer should not be confused with the various non-Microsoft enterprises (such as terraserver.com) that have nothing to do Microsoft even though they have managed to obtain confusingly similar domain names.

TerraServer may be accessed either via an interactive web site visited by users, or as a web service to which applications like Manifold may connect to automatically fetch desired images.

- This topic shows how to work with images obtained using the interactive TerraServer website and how to manually assemble them into larger mosaics.
- The Linked Images from TerraServer topic describes how to use the built-in Manifold capability of automatically downloading and combining images from TerraServer.

For most purposes, using the built-in Manifold ability to fetch and assemble TerraServer images automatically is much more convenient than manually visiting the web site and then downloading and assembling images by hand.

 Nonetheless, we have provided this example using TerraServer because getting images from TerraServer is similar to how they may be Obtained From other sources. It also provides a set of images that are easily georegistered using the Assign Projection dialog so that the topic can teach the simple cut-and-paste methodology of making mosaics from georegistered images instead of getting bogged down into lots of discussion on how to georegister images.

The downloadable images provided by Terraserver are smaller than the original USGS DOQ images from which they are generated. To create an image that covers a larger area one must download several adjacent images from Terraserver and then mosaic the images together into a single, larger image.

This topic shows how to download two adjacent images and to mosaic them together. Although the specific steps illustrated are set forth for Terraserver, the general ideas and methods can be applied to piecing together images from other public access web sites from which images may be obtained.

As a general comment, there is not much to creating image mosaics in Manifold if the images involved are correctly georegistered. Simply copy and paste images together to make a larger image. Most time in such projects is usually spent finding the images, downloading them and georegistering them. The actual assembly of multiple images into a single larger image is trivial.

Step 1: Download the images

Microsoft's TerraServer site allows users to download the images displayed in .jpg format. It also allows download of a “world file” for each image. The world file provides partial projection information for the image.

Visit http://terraserver-usa.com/ and locate the area of interest (this URL was active at the time this topic was written). For this example we will zoom into Palo Alto, California and use a USGS aerial photograph taken on October 30,1991. We zoom into the photograph to four-meter resolution for our area of interest. A screen shot showing part of the Terraserver web page is shown below. The arrows have been added to show three prominent landmarks.
We download the image seen above (using a menu on the web page that is not visible in the screenshot) as `palo_alto.jpg` and we also download the associated world file using the name `palo_alto.jpgw`.

Note the features shown by arrows in the illustration. We would like to download an adjacent image showing the region just below the region seen above. To do so, we pan the image downward by clicking on the "South" arrow at the bottom edge of the image on the Terraserver page.
Terraserver will scroll down one image at the current zoom level and show the display seen above. The same three landmarks are marked again with arrows. Note that when Terraserver scrolls to a new adjacent image it leaves some overlap with the previous image. We download this second image as `palo_alto2.jpg` and the associated world file as `palo_alto2.jpgw`.

If we take a look at our downloaded files with Windows Explorer we see that we have downloaded four files: two `.jpg` image files and two `.jpgw` world files. World files are simple text files that may be opened with Notepad.

**Step 2: Import and georegister the images**

Unfortunately, world files are obsolete in GIS usage because they do not encode the projection used. They simply provide certain local projection values and leave it up to the user to manually specify the actual projection in use.

If we opened up one of the world files in Notepad we would see the following for the `palo_alto.jpgw`:

```
8.000000
0.000000
0.000000
-8.000000
574400.000000
4147200.000000
```

...and this for `palo_alto2.jpgw`:

```
8.000000
0.000000
0.000000
-8.000000
574400.000000
4145600.000000
```

... That's all! No mention of any projection used, just the local projection parameters for whatever mystery projection was used. To find the projection used we will have to drill into the documentation posted in obscure places on the TerraServer site and then use the information we find to manually specify the projection information for each image within Manifold.

This is a phenomenally inconvenient way of doing things, which is why the GIS industry evolved the GeoTIFF standard that is now used by over 160 different companies instead of world files. However, Microsoft's TerraServer still uses world files. One suspects they do so because using GeoTIFF would force them to provide the images in TIF format while using world files allows them to serve images in JPG format and thus simplify their download process to simply saving whatever image has already been served to a web browser in JPG.

Be that as it may, if we want to use Terraserver images for free we must deal with world files. Lucky for us, Manifold is a powerful enough tool to cut through the hassles associated with world files. When Manifold imports an image it automatically looks in the folder from which the image was imported to see if there is a similarly-named file that uses one of the typical world file extensions. If such a world file exists, Manifold will extract the information it contains for use with local projection values. In this case, Manifold will find a `jpgw` file for each `jpg` file that contains the world file numbers. See the Importing Images topic for more information.
We import the `palo_alto.jpg` and `palo_alto2.jpg` image using File - Import - Image. If we open the Palo_alto image we see that Manifold has correctly imported the .jpg image we downloaded from the web site.
If we open the Palo_alto2 image we see that it too has been correctly imported.

In both cases, because the world files do not specify projections Manifold has imported both images using Orthographic projection. Because the use of Orthographic is a tip-off to Manifold that the image file format from which the images were imported does not provide full projection information, Manifold has raised an info bar on both images to warn us to verify the projection.

We can verify the projection and georegister the images at the same time, by clicking the info bar to launch the Edit - Assign Projection dialog for the image. We will have to manually specify the projection to be used since the world files do not do so.

The Assign Projection dialog is used to verify the projection if no changes need to be made or, if changes must be made, this dialog is used to specify the correct projection and parameters. We first click on the info bar for the Palo_alto image.

As seen above the dialog shows that the image has been imported using the default Orthographic projection used when no projection is cited by the format used. However, the Local offset and Local scale values have been correctly set to the values contained in the world file.

Check the Preserve local values box so that the local values imported from the world file will be retained when we change the projection to the correct projection.
We now must specify the correct projection. In this case, it is **Universal Transverse Mercator - Zone 10 (N)** and the datum to use is the **North American 1983**. See the notes at the end of this topic to learn how we know this magic information.

Hunt through the hierarchical list of projections in the dialog to find **Universal Transverse Mercator - Zone 10 (N)** and then choose this projection by clicking on it. Choosing a new projection would normally reset the values in the Local offset and Local scale boxes to the defaults for the new projection. Since we have checked the Preserve local values box the local values imported from the world file will not be changed.

After we press OK the image is georegistered. There will be no apparent change in the appearance of the images since georegistration in this case is simply a matter of telling Manifold how to interpret the coordinates it already has.

Repeat the above procedure to verify and georegister the **Palo_alto2** image as well by using the Edit - Assign Projection dialog.

**Step 3: Copy and paste to assemble a larger image**

Now that the images are georegistered the rest is easy.

Open the **Palo_alto** image. This is the "upper" image. Also open the **Palo_alto2** image. This is the "lower" image that overlaps the **Palo_alto** image along the upper edge.
With the focus on Palto, press CTRL-A or choose Edit - Select All to select all of the pixels. Press CTRL-C or Edit - Copy to copy all of the pixels.

Click on the Palto image to move the focus there and press CTRL-V or choose Edit - Paste to paste the copied pixels into this image. Manifold will ask if we want the image enlarged since most of the pixels added by the pasted image fall outside the original Palto image. Press Yes.
The result is that the *Palo_alto* image has been extended downward by the pasting of the pixels copied from the *Palo_alto2* image. The pasted pixels remain selected after the *Paste* operation.
If we click **Select None** we can see that the two images have been assembled into a single, larger image.
Zooming into the image we see that the alignment is excellent. The green lines mark the boundary between the original image and the pasted pixels from the other image. When TerraServer images originating in the same USGS DOQ are pasted together the boundaries are usually invisible.

**Notes**

This procedure uses two images that are vertically adjacent. The same process works when more images than just two are assembled. Import the images, georegister each using the **Edit - Assign Projection** dialog and then copy and paste them together into a single large image. When working with many images or large images, use a machine with lots of RAM, fast processor and adequate free space on disk.

How on Earth did we know that the correct projection to use was UTM zone 10? This resulted from some detective work and from some pure guesswork based on previous experience with USGS data sets. The TerraServer website provides only the following remarks in a discussion of the USGS "DOQ" images from which the TerraServer images are derived.

"The standard DOQ from the U.S. Geological Survey is a black-and-white (gray-scale) or color-infrared image covering 3.75 minutes of latitude by 3.75 minutes of longitude. Thus, four such photos can be combined in a mosaic to cover the area represented by a standard USGS 7.5-minute, 1:24,000-scale topographic map. Using mosaics is easier because images overlap. The DOQ's are referenced to the North American Datum of 1983 and use the Universal Transverse Mercator projection."

The datum in use is obviously the **North American 1983** datum, most likely that for CONUS (meaning the **Continental US**, since that's where Palo Alto is located), but the projection is not as explicit as we would like. The Universal Transverse Mercator projection is not one projection but rather a family of over one hundred projections. Saying an image is in UTM without saying which specific UTM zone was employed is somewhat like saying that an address is located at a postal code without specifying which postal code. It's possible to do some detective work to figure out what postal code is involved for a given street and city but it would be nice to simply have the postal code specified.
It's a reasonable guess that if the DOQs from which the TerraServer images are generated are in some UTM projection than the TerraServer images are also in the same UTM projection. We can also infer that if the DOQs are arranged so that they tile standard USGS 7.5-minute, 1:24,000-scale topographic maps then the DOQs use the same UTM zone employed in USGS 7.5-minute topo maps. If we know the USGS topo map in which the TerraServer image is located, we know the UTM zone used for that image.

The Manifold CD contains a .map file called USGS 24K Index.map that provides an index to USGS 7.5-minute, 1:24,000-scale topographic maps covering the US. We can open this map and based on our knowledge of where Palo Alto is located (it is located conveniently between Mountain View and Menlo Park) we can see that Palo Alto, California, is contained within the Palo Alto USGS topo map. If we didn't know where our images were located we would have to do some additional detective work, perhaps consulting an online atlas or other resources to figure out in which USGS topo map they reside. Using Manifold, we might add a layer of place names to the USGS 24K index map. For the images used in this example the task is easy but in cases where TerraServer images are in rural areas near the edges of topo maps we may have to work harder to figure out in which topo map our images are located.

Once we know the topo map in which our images are located we know the UTM zone they are in. Somewhere on Internet there is probably a list that specifies what UTM zone is used for each USGS topo map. However, there is a faster way to find out what UTM zone our images are in than to hunt all over Internet for some magic list of USGS topo maps and the UTM zone used for each.

We can simply go to the USGS web site and download a 1:24K scale Palo Alto map in SDTS format (USGS calls it an "SDTS DLG" map). We import it into Manifold, open it and look in the Edit - Assign Projection dialog to see what UTM zone is used. Because SDTS format correctly saves full projection information the UTM zone used will be correctly stated. We will see from the dialog that it is UTM zone 10.

![Map of Palo Alto](image1)

It's usually a good idea to download the USGS 1:24K SDTS DLG drawings for the region covered by the images being pasted together. The transportation / roads drawings from the SDTS map can be shown together with the images in a map to verify that georegistration has been correctly accomplished, as seen above.

One other use of copy and paste is to eliminate the small USGS logo that is embedded in each image downloaded from Terraserver. By pasting an adjacent image to the South the region of overlap will cover up the USGS logo with non-logo pixels. This effect can be seen in the example above where the USGS logo in the lower right of the Palo_alto image was covered up by overlapping pixels pasted from the Palo_alto2 image. The image can then be cropped back to the original size and the USGS logo will be gone. The images provided by USGS
are all in the public domain and do not need to be branded with the USGS logo if the user does not so desire. The original USGS DOQ images do not include small logos.

See Also

Linked Images from TerraServer
Active Columns using VBScript

This topic adds two active columns to the Products table from the Nwind.mdb sample database.

### Products

<table>
<thead>
<tr>
<th>Category ID</th>
<th>Product Name</th>
<th>Unit Price</th>
<th>Units In Stock</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Cha</td>
<td>$18.00</td>
<td>39</td>
</tr>
<tr>
<td>1</td>
<td>Chang</td>
<td>$19.00</td>
<td>17</td>
</tr>
<tr>
<td>2</td>
<td>Aniseed Syrup</td>
<td>$10.00</td>
<td>13</td>
</tr>
<tr>
<td>2</td>
<td>Chef Antoni's</td>
<td>$22.00</td>
<td>53</td>
</tr>
<tr>
<td>2</td>
<td>Chef Antoni's</td>
<td>$21.25</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>Grandma's Boy</td>
<td>$25.00</td>
<td>120</td>
</tr>
<tr>
<td>7</td>
<td>Uncle Bob's Or...</td>
<td>$30.00</td>
<td>15</td>
</tr>
<tr>
<td>2</td>
<td>Northwoods C...</td>
<td>$40.00</td>
<td>6</td>
</tr>
</tbody>
</table>

The Products table has fields for Category ID, Product Name, Unit Price, Units in Stock and various other columns not shown in the illustration above. Each product in the table has a Product ID. We will add two active columns that provide the category name for each product as well as the average price at which the product was sold.

The category name will be found by looking up the Category ID for the product in the Categories table.

### Categories

<table>
<thead>
<tr>
<th>Category ID</th>
<th>Category Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Beverages</td>
<td>Soft drinks, coffees, ...</td>
</tr>
<tr>
<td>2</td>
<td>Condiments</td>
<td>Sweet and savory sa...</td>
</tr>
<tr>
<td>3</td>
<td>Confections</td>
<td>Desserts, candies, s...</td>
</tr>
<tr>
<td>4</td>
<td>Dairy Products</td>
<td>Cheeses</td>
</tr>
<tr>
<td>5</td>
<td>Grains/Cereals</td>
<td>Breads, crackers, pas...</td>
</tr>
<tr>
<td>6</td>
<td>Meat/Poultry</td>
<td>Prepared meats</td>
</tr>
<tr>
<td>7</td>
<td>Produce</td>
<td>Dried fruit and bean c...</td>
</tr>
<tr>
<td>8</td>
<td>Seafood</td>
<td>Seaweed and fish</td>
</tr>
</tbody>
</table>

The Categories table provides a Category Name for each Category ID.

The average sales price will be found by looking up all instances of the Product ID for the product in the Order Details table.

### Order Details

<table>
<thead>
<tr>
<th>Order ID</th>
<th>Product ID</th>
<th>Unit Price</th>
<th>Quant</th>
</tr>
</thead>
<tbody>
<tr>
<td>10000</td>
<td>4</td>
<td>$22.00</td>
<td>1</td>
</tr>
<tr>
<td>10000</td>
<td>5</td>
<td>$21.35</td>
<td>1</td>
</tr>
<tr>
<td>10000</td>
<td>17</td>
<td>$27.00</td>
<td>4</td>
</tr>
<tr>
<td>10003</td>
<td>18</td>
<td>$43.70</td>
<td>12</td>
</tr>
<tr>
<td>10004</td>
<td>29</td>
<td>$36.00</td>
<td>35</td>
</tr>
<tr>
<td>10004</td>
<td>63</td>
<td>$30.70</td>
<td>6</td>
</tr>
<tr>
<td>10005</td>
<td>62</td>
<td>$34.00</td>
<td>5</td>
</tr>
</tbody>
</table>

This table lists all products in all orders. By finding every occurrence of a particular product we can see what the unit price was that was actually applied for that product in the order.

**Step 1: Add Active Column**

Open the Products table and right click onto any of the column heads and choose Add - Active Column.
In the **Add Active Column** dialog.  Provide the name of the function for the **Category Name** column and other parameters as shown above.  The function name used is the name of the function that we will write into the script in the next step.  Press **OK** and the **Products Script** for the **Products table** will be created and opened for editing.

**Step 2: Enter the script**

The script component is set up to use VBScript by default.  To save time, we will enter the functions for both active columns into the script component at this time:

```
' -- determines average price for product using [Order Details] table
Function AveragePrice

    ' -- obtain product ID
    product = Record.Data("Product ID")

    ' -- obtain details of orders for product
    Set details = ComponentSet(ComponentSet.ItemByName("Order Details"))
    Set detailsForProduct = details.RecordSet.EqualTo("Product ID", product)

    AveragePrice = detailsForProduct.Average("Unit Price")

End Function

' -- looks up product category name within [Categories] table
Function CategoryName

    ' -- obtain category ID
    category = Record.Data("Category ID")

    ' -- obtain category name from Categories table
    Set categories = ComponentSet(ComponentSet.ItemByName("Categories"))
    Set catRecords = categories.RecordSet
catRecordIndex = catRecords.ItemByValue("Category ID", category)
    If catRecordIndex < 0 Then
        CategoryName = ""
    Else
        CategoryName = catRecords(catRecordIndex).Data("Category Name")
    End If

End Function
```
**Step 3: Add second Active Column**

Add the second active column by right clicking onto any column head in the table and choosing **Add - Active Column**.

We add the active column called **Average Price**. The function called is the **AveragePrice** function written into the script in the previous step.

After creating the column we right click on the **Average Price** column and choose **Format** and set the formatting to use $ currency formatting.

The result of the above is that two new columns appear in our table which are computed using the functions in the script.

**See Also**
See the Scripts topic for information on scripting. See the Active Columns topic for more on active columns.
Active Columns using JScript

This topic adds three active columns to the Employees table from the Nwind.mdb sample database.

<table>
<thead>
<tr>
<th>F..</th>
<th>Last Na..</th>
<th>First N..</th>
<th>Birth Date</th>
<th>Hire Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Fuller</td>
<td>Andrew</td>
<td>2/19/1942</td>
<td>7/12/1991</td>
</tr>
<tr>
<td>4</td>
<td>Peacock</td>
<td>Margaret</td>
<td>9/19/1957</td>
<td>3/30/1992</td>
</tr>
<tr>
<td>5</td>
<td>Buchanan</td>
<td>Steven</td>
<td>3/4/1955</td>
<td>9/12/1992</td>
</tr>
</tbody>
</table>

The Employees table has fields for Employee ID, Last Name, First Name, Birth Date and Hire Date. We will add three active columns that provide the full name of the employee, the age of the employee and the years since the employee was hired.

**Step 1: Add Active Column**

Open the Employees table and right click onto any of the column heads and choose Add - Active Column.

In the Add Active Column dialog, provide the name of the function for the Full Name column and other parameters as shown above, including choosing JScript as the language to use. The function name used is the name of the function that we will write into the script in the next step. Press OK and the Employees Script for the Employees table will be created and opened for editing.

**Step 2: Enter the script**

To save time, we will enter the functions for all three active columns into the script component at this time:

```javascript
// compose full name from [Last Name] and [First Name]
function fullName() {
    return Record.Data("Last Name") + ", " + Record.Data("First Name");
}

// return number of years since given date to current moment
function years(arg) {
    today = new Date();
    todayYrs = today.getFullYear();
    query = new Date(arg);
    queryYrs = query.getFullYear();
}
```
// return number of years since [Birth Date] to current moment
function yearsOld() {
    return years(Record.Data("Birth Date"));
}

// return number of years since [Hire Date] to current moment
function yearsWorking() {
    return years(Record.Data("Hire Date"));
}

**Step 3: Add remaining Active Columns**

Add the remaining active columns by right clicking onto any column head in the table and choosing Add - Active Column.

First we add the active column called *Years Old*. This active column uses the `yearsOld` function written into the script in the previous step.

Next we add the active column called *Years Working*. This active column uses the `yearsWorking` function written into the script in the previous step.
The result of the above is that three new columns appear in our table which are computed using the functions in the script. Note that two of the functions use the `years` function defined in the script.

**See Also**

See the Scripts topic for information on scripting. See the Active Columns topic for more on active columns.
Show Area of a Parcel in Acres

Manifold features may be easily combined to create a virtually unlimited range of capabilities. In this example we will combine some simple features to report in acres (or hectares) the surface area of a parcel.

For this example we will use the ZCTA Zip Codes sample project.map file. This shows ZIP codes (postal codes) in the US as areas. We will find the acreage of areas.

Step 1: Make sure the drawing is projected

Load the ZCTA Zip Codes.map sample projects. Open the Zt11_District of Columbia drawing.

To compute areas accurately, we must use the drawing in projected form. Using the Edit - Assign Projection dialog we verify the initial projection of the drawing as Latitude / Longitude. Manifold requires us to do this before we can change the projection of the drawing.

Next, we use the Edit - Change Projection dialog to change the projection to Orthographic, making sure to use the Suggest button to suggest appropriate parameters. Projections will use either meters or feet as units of measure in their coordinate systems. Note that the Orthographic projection uses meters as units of measure in its coordinate system.

Any projection which provides accurate measurement of areas would work just as well. See the Guide to Selecting Map Projections topic for a discussion of different projections.
The projected drawing has a slightly different appearance. This is how the District of Columbia would appear in real life if viewed from an aircraft directly above the District.

**Step 2: Show areas in the drawing’s table**

In the **Miscellaneous** section of the **Tools - Options** dialog make sure the **Use English measurement units** box is checked.

Open the drawing's table. In the **View - Columns** dialog, enable the **ZIPCODE** and **Area (I)** columns for display.

<table>
<thead>
<tr>
<th>ZIPCODE</th>
<th>Area (I)</th>
</tr>
</thead>
<tbody>
<tr>
<td>20012</td>
<td>69020717.2976609</td>
</tr>
<tr>
<td>20015</td>
<td>92769003.6535265</td>
</tr>
<tr>
<td>20011</td>
<td>130659981.292517</td>
</tr>
<tr>
<td>20008</td>
<td>84789312.0459514</td>
</tr>
<tr>
<td>20016</td>
<td>125734696.95095</td>
</tr>
<tr>
<td>20017</td>
<td>63914572.4810944</td>
</tr>
<tr>
<td>20018</td>
<td>87222502.3952926</td>
</tr>
<tr>
<td>20010</td>
<td>43557760.6244952</td>
</tr>
<tr>
<td>20011</td>
<td>191081661.803211</td>
</tr>
<tr>
<td>20009</td>
<td>37223355.8283784</td>
</tr>
<tr>
<td>20001</td>
<td>72836196.4501965</td>
</tr>
<tr>
<td>20007</td>
<td>94927481.9962765</td>
</tr>
<tr>
<td>20002</td>
<td>141327436.870931</td>
</tr>
</tbody>
</table>

The table now reports the area of each object in the drawing in the **Area (I)** **intrinsic field** column, using square feet as the units of measure. It uses square feet because we checked the **Use English measurement units** box in the **Tools - Options** dialog; if this box were not checked, then the **Area (I)** intrinsic field would report the areas as square meters since the Orthographic projection uses meters as units of measure.

At this point we already have our results. We can select any area in the drawing and the row corresponding to that area will also be selected in the table. We can then see the surface area of that area object in square feet listed in the **Area (I)** column. Divide that value by 43560 to get the surface area in acres (one acre = 43560 square feet).

From this point onward we can do a few additional steps to increase the user friendliness of finding areas. For example, we can make the selection more obvious in the table.

With the focus on the table, press in the **Selection Filter** button. This will cause the table to show only selected records.

4121
If no area is selected the table will appear empty.

Using touch select, select an area in the drawing.

The record for the selected area will appear in the table. With the Selection Filter button engaged, it is easy to resize the table window so that it occupies a small portion of the display, allowing a larger drawing window. We can then click at will on different areas in the drawing window to select them and immediately see their surface area in square feet.

**Step 3: Automatically compute acreage**

It would be more convenient if Manifold calculated the surface area for us in acres. This is easy: we can use an active column to automatically convert square feet into acres.
Right click on any of the table's column headings and choose **Add - Active Column**. In the resulting dialog we specify the characteristics for the new active column. We will use **Acres** for the name of the new column. It will be a **Floating-point (double)** type that will be computed **on user request**. Press **OK**.

Creating an active column means creating a script that implements the functionality of the column. To make this easier, Manifold helpfully creates a new script populated with some text to use as a starting point.

We edit the starting text as seen above.

```
Function Func
    Func = Nothing ' Record.Data("Column")
End Function
```

```
Function Func
    Func = Record.Data("Area (I)") / 43560
End Function
```
The function we have created tells Manifold to take the value from the *Area (ft)* column for each record and to divide it by 43560. This will convert square feet into areas. If we preferred some other measure, such as hectares or square miles or square kilometers we could have just as easily used the conversion factor to those from square feet instead of 43560.

Back in the table, we right click on the *Acres* column head and choose *Recompute* (note that we right click on the column head). The *Acres* column now reports the surface area of that area object in acres.

When we created the active column we could have instructed Manifold to recomput the column automatically by choosing a different setting in the *Compute* box in the *Add Active Column* dialog. That is not usually done in applications like this because area objects in drawings are not usually continuously changed, so there is no point to frequent recomputation of surface area. It’s enough to do it once, on demand, as necessary.

We now have a very useful arrangement. Every time we select a different area object in the drawing...
The table will immediately report the surface area in acres.

One very useful aspect of active columns is that they work just like any other table column. For example, now that we've defined an active column we can see the value of it for any object by simply double-clicking the object to call up the Object Fields dialog.

We have to press in the r/o (Show Read-Only) button to show the Acres column because it is a read-only column. Together with viewing selected records in the table this gives us a second way of seeing the acreage of any area object in the drawing.

**Step 4: Create a ViewBot**

Yet a third way of displaying acreage is to create a ViewBot that shows the total acreage of all selected areas. This is easy to do.

Choose View - Panes - ViewBots to turn on the ViewBots pane.
Choose New ViewBot to add a ViewBot to the pane.

In the Add ViewBot dialog we choose the [Selection] as the scope, the Acres column as the source of our data and Sum as the operation. This tells the ViewBot to report the sum of all values in the Acres column for all selected records. The Caption is whatever text line we want to remind us of what the ViewBot is telling us. We choose to Autorefresh to make sure the ViewBot is always up to date when the selection changes. Press OK.

The ViewBot immediately reports the total acreage for the selected area object.
If we click on a different area in the object to select it...

...the ViewBot immediately reports the acreage for that object.

What is very useful about our new ViewBot is that we can select several area objects (using Add to Selection)...
...and the ViewBot will report the sum of the acreages in all the selected area objects. This allows very rapid interaction with the drawing. We can very rapidly add or remove areas from the selection to see exactly how they sum up in acreage. In fact, we can easily save various selections as saved selections, create new selections and so on to use this capability for a sort of manual territory balancing or other interactive work.

If we have the table window still open, note that it continues to show selected records. We can use the ViewBot to see the sum of the acreages and the table to see values for individual records. Outstanding!

ViewBots are such a useful tool that we can add a few more to the ViewBots pane.
We will add a ViewBot that reports the largest parcel in the selection.

We will add one to report the smallest parcel in the selection.

And finally we would like to know the average acreage of parcels in the selection.
We now have a wonderfully interactive display that allows us to select whatever we like and immediately see the total acreage as well as the largest, smallest and average parcel acreages. Outstanding!

Notes

The ViewBot and table will continue to report values for selected items no matter how selection is accomplished. For example, we can select objects using queries or via the Query Toolbar or by other methods. Whatever is selected by whatever means, the ViewBot and table will faithfully report the acreage.

See Also

Intrinsic Fields in Tables
Active Columns
ViewBots
Selection
Query Toolbar
Selecting Objects with Queries
Exporting KML to Google Earth

This is a long topic because it shows every detail for beginners. An experienced Manifold user could do this topic in about five minutes.

Google users may find it odd that this tutorial is written using Microsoft Virtual Earth to provide imagery backgrounds instead of Google Earth. The substitution of Microsoft Virtual Earth images for Google Earth images was done at Google's request to ensure that the numerous images that appear in this topic when it is posted to the manifold.net website do not show what Google regards as copyrighted imagery. See the note at the end of this topic for more information.

The Tutorial

This topic shows how to open the base map project and how to zoom into a location. We then add points showing the locations of our favorite restaurants and we export the points to Google Earth. The topic assumes that the StarterPack.zip file has been downloaded and installed, that the computer has an Internet connection and that Google Earth has also been installed on the computer.

The StarterPack.zip file contains a starter .map project as well as DLLs for open source image server drivers, including a Microsoft Virtual Earth image server driver. It should be unzipped into the Manifold System installation folder (C:\Program Files\Manifold System by default). If you have not downloaded and installed the StarterPack.zip file this tutorial will not make any sense because the US Base Map.map project file it mentions will not be available to you.

The StarterPack.zip archive must be downloaded from links on the manifold.net web site or from whatever web site you prefer to get your Manifold open source modules.

Note: The usage of Microsoft Virtual Earth image servers, as seen in this tutorial, requires installation of open source software written using the Manifold Image Server Interface. There are many third parties providing free image server modules for image servers like Microsoft Virtual Earth, Yahoo! Maps and others. You can even write your own! Manifold does not supply or support such third party software, but it is really, really cool to know that you can do such things with Manifold and that there is an active open source community supporting Manifold.

The StarterPack.zip archive provides open source software that was contributed to the Manifold user community. The base map was created and contributed by Manifold users as well. The amazing thing is that you don't need to know anything about Manifold internals to do such stuff, nor do you have to have more than ordinary user skills to create things like the base map.

Step 1: Open the base map project

In Windows Explorer, browse into the Manifold System installation folder (C:\Program Files\Manifold System by default) and double-click on the US Base Map.map project file.

Manifold will launch. If this is the first time you have launched Manifold, an Activation dialog will pop up. You must enter at least the Manifold System serial number that was sent to you with your license. If it is longer than 30 days from the date the serial number was issued, you must activate your license as well. See the Activation Keys and Serial Numbers topic in Help.
Manifold launches with a Project pane open on the right that lists all components in that project. There are no windows open in the workspace. The main menu appears at the top and toolbars appear just below the main menu. The illustration above has been reduced.

If we take a closer look at the project pane, we see that each component has an icon next to it depending on what it is. Drawings have a little pencil in the icon. Images have blue lines in the icon, like raster lines on a monitor. Items that linked in from some external data source, like a database or a web server have a little yellow database cylinder in the icon. Comments have a book.

All these items can participate as layers in a map. There is only one map in this project, called "Map" so that beginners don't have to guess what it is. Double click on Map to open the map.
Example: The “Local” tool tip that pops up for many items in the project pane is Manifold's way of telling you that item is stored locally, inside the project .map file. Items also can be linked into a project from elsewhere, just like the linked images from the Microsoft image servers. In that case, the data for those items is not stored inside the project file.

When the map pops open Manifold automatically pops open a default set of toolbars for use with that map and adjusts the main menu with appropriate entries. A map window can be resized and moved around in the Manifold workspace. We can pop open as many windows as we like in the workspace. If we like, we can right click on the Map item in the project pane and choose Open in New Window and Manifold will merrily pop open that same item in a second window for us. That second window can be panned and zoomed independently of the first.

It's not easy to show the entire Manifold workspace in a tutorial like this without using reduced images that are difficult to read. From here on, this tutorial will not show the entire workspace but will just show the main windows as needed in full size illustrations. Let's take a closer look at the map window that we've just opened.
There, that's better! The beautiful background image we see has been browsed automatically for us from Microsoft Virtual Earth. A hallmark of Virtual Earth is the superior, smooth, beautifully realistic quality of the image even at larger scales, as though we are actually hovering over the planet and viewing the scene in real life. The equivalent Google image, which we are not allowed to show you, appears as a patchwork mosaic of annoyingly varied tones without the sense of true reality conveyed by Virtual Earth.

A map window shows a stack of other items as layers. The layer tabs at the bottom of the map window show what layers are in the map and which layers are turned on for display. A layer tab with disabled text means that layer is not turned on for display. In the illustration above, the Drawing layer and the USA layer are turned on.

The Drawing layer is empty (for now) and the USA layer just shows outlines of states in the US using green lines.

The Drawing layer tab has a white background. This is the active layer, the one in which any editing commands will take effect. Click on a layer tab to make that layer the active layer. Double-click a layer tab to turn it off or on for display.

In this example we will add a few points to the Drawing layer and then we will export the Drawing layer as a Google Earth KML file. All the other layers in the map are just there in case we want to use them as reference points or to otherwise help us draw whatever we want into the Drawing layer.

There are too many layers in this map to show all of them at once as layer tabs, so we can use the scroll buttons in the lower left of the map window to scroll through the layer tabs. There is one more layer that is turned on for display, the Virtual Earth Satellite Image layer. This layer automatically fetches an image from the Microsoft Virtual Earth servers at the required zoom and resolution.

**Step 2: Open the layers pane**

The layer tabs are very handy for simple maps, but when maps have very many layers it is usually easier to work with layers by opening the Layers pane.
We open the Layers pane by choosing View - Panes - Layers from the main menu.

The layers pane pops open by default underneath the project pane. It will be too small to show all the layers initially, so we can resize the pane by clicking and dragging on the boundary between the layers pane and the project pane as seen above.
The layers pane shows all the layers in whatever window has the focus (that is, the last window we clicked on) in the workspace. A checked box indicates that layer is turned on for display. We can turn layers off and on by un-checking or checking their boxes.

The map window now shows the **Virtual Earth Satellite Image** layer as the lowest layer, "behind" all the other layers.
Un-checking the **Virtual Earth Satellite Image** layer box in the layers pane will turn off that layer.

The result in the map window is the **USA** layer seen by itself against the default white **Background** layer.
Back in the layers pane, let's check the **Virtual Earth Street Map Image** to see what that is all about.
It appears in the map window. Note that the USA layer is still turned on. The USA layer just contains lines with "empty space" between the lines. Any layer below the USA lines will show through the empty space.

The Microsoft image servers basically have two modes of operation as a web service. They can show street maps like the Virtual Earth Street Map Image layer or they can show mosaics of satellite photos like the Virtual Earth Satellite Image layer. The starter base map contains both as layers because sometimes it is more convenient to use a satellite image as a background and some times we would prefer to use a street map image.

**Step 3: Zoom into a desired location**

In the layers pane, please un-check the Virtual Earth Street Map Image layer checkbox and check the Virtual Earth Satellite Image layer checkbox so that the satellite image layer is displayed. This step is not illustrated.

Let's now zoom into a region of interest. For that, we will use the Zoom Box tool.
Click on the **Zoom Box** tool to make it the command mode for the mouse. Until we unclick the tool, the mouse will be in **Zoom Box** mode when we use it in the map window.

We're going to go to Palo Alto, California, so we'll click and drag open a zoom box around the green squiggles that indicate San Francisco Bay, on the California coast. [Readers who don't know which state is California without a label should be ashamed, deeply ashamed... It's not like it's a hard state to remember, like one of those square states in the middle that all look alike! ]
Manifold zooms in for us and automatically fetches higher resolution imagery from the Virtual Earth server. Let's zoom farther into the San Francisco Bay region.

Excellent. Let's do that one more time, using the **Zoom Box** about where we think Palo Alto is located.
Manifold keeps loading ever more detailed imagery from the Microsoft Virtual Earth server. The problem now is that we’re not sure exactly what we are looking at. There’s a bridge there, but darned if we’re not exactly sure where to zoom now. Now is a good time to switch to using the street map image.
In the layers pane we un-check the Virtual Earth Satellite Image layer and we check the Virtual Earth Street Map Image layer.

In the map window we can see we zoomed into a region too far northwest. We've zoomed into San Mateo and Hillsborough, and not Palo Alto. That's easy to fix.
Click on the **Grabber** tool to make that the command mode for the mouse.

In the map window, click in the lower right corner and then drag to the upper left and release. We are telling Manifold to grab the portion of the view from the lower right corner and move it up to the upper left part of the window. This scrolls the entire window in the desired direction.
Manifold obliges, scrolling the window as we wanted. There's Palo Alto near the center of the display.

We click back onto the **Zoom Box** tool and then zoom into the Palo Alto section of the map.
We keep zooming until we reach the downtown portion of Palo Alto where dozens of restaurants cater to Silicon Valley yuppies and Stanford University students and faculty. Stanford is about a half mile out of view to the lower left in the above map.

**Step 4: Add a point**

We are now zoomed into the map and are ready to add some points that show the locations of our favorite restaurants.

Before we continue, if you have been fooling around with the map outside of the sequence of steps in this tutorial, click on the **Drawing** layer tab to make sure it is the active layer. We will be adding points to the **Drawing** layer.

Next, a little housekeeping: We are going to turn on **Instant Data** mode so that when we click to add a point we can easily also add some data for that point.

We do that in the main menu by clicking **Edit - Instant Data**. This puts a check mark next to the **Instant Data** entry on the menu to let us know that **Instant Data** mode has been turned on. We'll see what this mode does in an instant.
To begin adding points we click the **Insert Point** tool in the drawing toolbar. This puts the mouse into **Insert Point** command mode.

Position the cursor at the corner of the intersection of University Avenue and High Street as seen above and click.

If we didn’t have **Instant Data** mode on, a dot would have instantly appeared. If we just wanted to do something really simple and click to create points we could do that all day without using **Instant Data** mode. But because we’re using Manifold and not some retarded “sketch” program, we want to do more than just click and create dots. We want those dots to be connected with some data, like the names of restaurants, that Google Earth can then use as a label or a description. Using **Instant Data** makes that especially easy to do.

With **Instant Data** mode turned on, a dot appears and a dialog pops open that allows us to enter data for the point. The dialog pops open with a field already opened and the text cursor in the box ready for us to enter text.

All drawings in Manifold have a table that’s connected to that drawing. Every object in a Manifold drawing, that is every point, every line and every area, has a record in the drawing’s table. Tables can have all kinds of data fields in them. It’s a cool way of having data associated with things we see in a drawing, like points.

The drawing called "Drawing" in this project has only two fields in its table, as that is how it was created when the starter map project was made. These are called **Name** and **Description**. It takes only a second or two to add fields or otherwise change the structure of a table. For now, we’ll use the table as is.

With **Instant Data** mode turned on Manifold immediately sets us up to add data whenever we create a new object. In this case we clicked using the **Insert Point** tool to create a point, so Manifold pops open the dialog to allow us to enter the data we want associated with that point.
We enter the name **Miyake** for this point and we enter the URL for this restaurant into the description field. Press **Enter** or click **OK**.

A new point appears using gray default color and default size. That gray color is boring, so let's change the color to bright green.

We do that by clicking on the **Point Background** color well..
Examples

...and choosing a nice bright green color.

The point appears in bright green. This particular point style (shown in the Point Style well that's just to the right of the Point Background color well we changed) uses foreground color for the border of the circle and background color for the fill color. If we wanted the points to be bigger we could click the Point Size well and make the size larger than the default size of 3.

There are two interesting things to note about the above illustration: first, by default the format color settings apply throughout an entire layer. When we make the Point Background color bright green, all points in that Drawing layer will be bright green. There are ways of changing this so that different points have different colors but since this is a beginner's tutorial we won't get into that.

The second interesting thing is that Google's first Palo Alto offices were about 100 feet to the lower right of that green dot in the illustration above. What is now one of the most valuable businesses in the planet started in cramped, second floor offices behind a video rental store near that green dot. Even more interesting is that the original offices of Facebook are just across High Street from the green dot. Amazing! ...Over one hundred billion dollars in wealth created on either side of a small sushi restaurant in Palo Alto.

**Step 5: Add another point**

Let's add some more restaurants.
The mouse is still in **Insert Point** command mode, so we will click on the corner of Emerson Street and University Avenue.

**Instant Data** mode is still on, so the dialog pops open again. **Instant Data** repeats the last data entered and positions the cursor within the last entered box ready to go. This is a big help with repetitive data entry. For example, if a city employee were marking the locations of different types of city assets and marking all hydrants, it is a real time saver to not have to manually type "hydrant" thousands of times.
In this case, we're marking the location of a sushi restaurant that does not have a web site so we'll just enter **Japanese** as the description...

...and then enter the name of the restaurant in the **Name** field. Press **Enter** or **OK**.
Manifold creates a new point at the location clicked. Note that since formatting works by default for all objects in a drawing, all the points in this drawing layer will be green, use the same style and be the default point size of 3.

We can continue to add as many points as we like. The above procedure is very fast, so it's easy to add points. It's also easy to add URLs. We like to keep an Internet Explorer browser session open so that we can find the URLs for those restaurants that have a web page. We can then Copy the URL from the browser window with a CTRL-C and then switch into Manifold and Paste the URL into the Description box with a CTRL-V. Fast and easy!

**Step 6: Correcting errors**

Suppose we get a little too excited about all this and click to create a point in the wrong spot? There are two ways of dealing with this.

Consider the point shown above with a red arrow. We accidentally added it in the wrong location. The first way to remedy the error is to select the point and delete it and then create a new point in the correct location.

Choose the **Select Box** tool.

Click and drag to make a selection box around the point. The **Select Box** tool selects all objects that are entirely within the box.
The point is shown using red selection color for foreground color to show it is selected.

If we choose **Edit - Delete** in the main menu or press the **Delete** key the point is deleted. For that matter, we could also have chosen **Edit - Cut** or **CTRL-X** to use Windows **Cut** to delete the point and simultaneously copy it to the Windows Clipboard. (Yes, Manifold can copy objects into the Windows Clipboard. These can then be pasted as new drawings or pasted into other drawings.)

Suppose, however, that we’ve entered some really long text into the **Description** box and we don’t want to re-enter it. In that case, instead of deleting the point we can just move it to the desired location.

To do that, we first make sure the mouse is not in any command mode. For example, if we have the **Insert Points** toolbar button pushed in, we first click it out. When the mouse isn’t busy with some command mode it can be used for **smart editing** in Manifold, that is, a streamlined, simple, default editing regime.
**CTRL-ALT** click on the point to be moved. That is, while holding down the **CTRL** key and the **ALT** key, click on the point.

This selects the point for editing, and a square box editing handle appears at the point's location. The editing handle can be clicked and dragged, moving the object as well.

We can click and drag the point to the desired location.

After we release to move the point to a new location, it still stays selected for editing in case we want to do anything else with it.
Deselect it by clicking the Select None button. (There are many other shortcuts which we will not get into here).

The point appears in its new location at the corner of Ramona Street and Hamilton Avenue. Many Google employees will immediately recognize this as the location of the Osteria restaurant, a great spot for osso buco with a nice Chianti classico.

**Step 7: Preview our work before export**

So far we've added points using Virtual Earth street map imagery as a background. That's very cool and useful but it is not the satellite imagery used by default in either Virtual Earth or Google Earth.
If we show the entire map window again, we can see that several additional restaurant locations have been added. Each one also has had a name and description provided. Let’s see how these points appear with a satellite image background.

In the layers pane we uncheck the street map image layer and check the satellite image layer.

OK, that’s very cool. The green dots are harder to see with all that photographic imagery clutter (easy to fix, see below) but the screenshot above shows us how Manifold can preview what will appear in Google Earth by using satellite imagery from image servers, in this case using imagery from Microsoft.
This is also a chance for us to move points around using the photograph as a visual reference instead of the street map image. If a point appears, say, in a parking lot just next to a restaurant instead of on the building itself we could move it using the procedure given above.

This is one reason why it is so valuable to prepare KML files in Manifold, because you can see where the data is going to appear using either street map imagery or overhead aerial photos like the one seen above.

Now would be a good time to save the project using **File - Save** to some safe location.

**Step 8: Create a Google Earth File**

Actually creating a Google KML or KMZ file from our drawing is absurdly easy.

We right-click on the **Drawing** in the project pane and in the menu that pops up...

...we choose **Export**.
In the Export Drawing dialog that pops open we choose KML / KMZ Files in the Save as type box and we provide a file name. If the file name we provide ends in .kmz Manifold will create a KMZ file. If the file name ends in .kml Manifold will create a KML file. The two are the same thing except that KMZ is a compressed form of KML using “zip” style compression.

For this example, we create a KMZ file using the name sample restaurants.kmz. Press Save.

The Export KML File dialog pops open to allow us to choose which fields we want to use for Google Earth Name and Description fields.

The pull down boxes for the Name and Description fields show all the fields available in the drawing's table, including all sorts of weird internal fields that are used by Manifold experts. We ignore all those and simply choose the Name and Description fields we have used to store restaurant names and descriptions. The Drawing layer in the starter project uses these two names for the fields it contains to make life easier for beginners. Press OK.
If we pop out of Manifold and take a look with Windows Explorer, we see that Manifold has created a new file for us called **sample restaurants.kmz**. We can double-click this file to launch it in Google Earth.

Google Earth will fire up, do the "zoom in from space" thing and, Voila! There are our points. Note that Google Earth by default will take the contents of the **Name** field and use it to label each point. Note also that the imagery used by Google Earth is indeed just like the satellite imagery the image servers streamed into Manifold as the background for our map.
The Google Earth Places pane shows both the Name and the Description values for each point. We obviously didn't have a lot of URLs for the restaurants of interest. What's cool about this Places pane is that if you do have a URL as the description, clicking on that URL in the Places pane will launch it in a mini-browser in Google Earth.

There are many ways of using URLs. For example, suppose we snapped pictures of each restaurant and then uploaded them to a public photo viewing site. For each restaurant we could put the URL to the photo in our drawing's Description field and then it would have been uploaded to Google Earth so that people could click to see what the restaurant looks like. This would be especially useful if our business was real estate sales and we were marking locations of houses for sale as opposed to our favorite restaurants.

**Step 9: Useful Manifold Features**

There are thousands of Manifold features useful in various situations. Here are a handful out of those thousands.

**Opacity:** Suppose we don't like the visual clutter of the photographic image in our map but we still want to have it available as a visual guide in the background. That's easy to do by adjusting the opacity of that image layer.
In the layers pane we click the **Opacity** button to turn on the **Opacity** column in the pane. This shows the opacity (that is degree of transparency) for each layer. We double-click into the **Opacity** cell for the **Virtual Earth Satellite Image** layer and change it to **40%**.
The result is that the layer fades out as part of the white background layer shows through. This makes it easier to see the points.

**Labels:** Suppose we have trouble remembering which green dot is which restaurant and we would like to add labels in the Manifold map, one for each point that gives the **Name**. That's easy also:

1. We right-click on the **Drawing** component in the project pane and we choose **Create**.

2. In the **Create** menu that pops up we choose **Labels** to create labels.
In the Create Labels dialog that pops up we choose a name for the new Labels layer we are creating.

To create the text automatically for each label, we double-click on the Name field in the Columns pane. That adds it to the Text box to be used as the text for each label. Press OK.
This creates a new labels component, called **MyLabels** just as we wanted, in the project pane. It is shown indented under the **Drawing** drawing because it is created automatically using the data in that drawing's table. To add this layer to the map we drag and drop it into the map.

First, we click on the **Drawing** layer tab in the map to make it the active layer. Now, when we drag and drop the **MyLabels** layer from the project pane into the map it will appear just above the **Drawing** layer.

Using the mouse, click on the **MyLabels** item in the project pane and then drag the mouse cursor over into the map window and release.

The **MyLabels** layers pane appears in the map window as a new layer and the labels appear using default formatting, positioning and font. It's a big uglier than we would like.
For starters, we would like to change positioning so that instead of being on top of each point the labels are below and to the left. This is easy to do.

When a labels layer is the active layer in a map, Manifold configures the toolbars with controls for labels. We can change the alignment settings (seen above) from centered to appearing to the left and below.

That's better. However, it would be cool if we could adjust the formatting of the labels so that they use black text with a bit of white outline color for better legibility.

To do that we click on the Label Style well in the formatting toolbar.
We choose a style with a bit of background color.

We then click into the **Label Background** color well...

...and we change the background color to white.
The result is a subtle change but highly useful when labels are to appear before a background that varies in color from light to dark. Note that these label changes appear within the Manifold project. Google Earth uses its own fonts, so these changes are not exported into Google Earth. But they are nonetheless highly useful because if we want to manipulate sophisticated data within Manifold we will often want to use labels in sophisticated ways to help keep our bearings.

**Changing Data for Descriptions:** Suppose we would like to update the data for each point, say, by changing the description? That’s easy to do by editing the drawing’s table.

Double-click the drawing’s table to open it in a table window.
There’s the data we entered for each point. We can change a value by double-clicking into the cell.

Here we’ve changed the name of Pizza My Heart to Pizza A Go Go, the name under which they do business for deliveries. We’ve also changed the Description for that record to include a totally off the wall and useless reference to the manifold.net home page.

The changes take immediate effect in the Manifold project. For example, the label for that restaurant (which automatically uses text from the Name field) instantly changes to show the new name. Note, by the way, that Manifold by default will "clip" labels that get in the way of each other. This can be turned off if desired.
If we export the drawing to a kmz file and open that file in Google Earth, we see that the new name is now used for a label in Google Earth. Google Earth doesn’t like to clip labels, and will try to move them around to show as many as it can no matter what.

In the above case, Google Earth tries too hard to show all labels because it has moved the Sushi Tomo label to the right and the Pizza A Go Go label to the left when in reality the two points are on opposite sides of the street the other way around.

If we zoom farther into Google Earth so there is more room for labels then Google Earth will show the labels in correct position nearer to the actual points they label.
The description in Google Earth's Places pane has also changed. Note that Google Earth is smart enough to know a URL when it sees one, so we can embed URLs within other description text and still be able to launch the URL with a click.

Notes

Why Palo Alto in the example? That's where Stanford University is located and where Google was founded. The restaurants mentioned are all places our friends at Google know. It's also amusing that the surprisingly muscular new CloudMade startup which is challenging Google Maps by leveraging the community intelligence of the OpenStreetMaps project is right next door, in Menlo Park.

Isn't there a faster way of finding a place in Manifold than using Zoom Box? Sure. If you have the Geocoding Tools extension installed you can GoTo an address, city name, ZIP code, etc. Even without the Geocoding Tools extension you could select a city name using the selection toolbar and then jump to that selection. There are often very many ways of doing something in Manifold so that an experienced user can always pick a method that is just right for the job.

It's fun to export points, lines and areas and to extrude them into 3D shapes. Extruding areas is a great way to make simple "buildings" that will appear in 3D when Google Earth is viewed with an inclined view. See the Export Drawing - KML, KMZ topic for an illustration.

Legal Note

This topic originally used Google Earth imagery because our writers felt that people interested in exporting KML to Google Earth would probably feel most comfortable with Google Earth imagery as the background layers used to create such KML. Makes sense, right?

However, Google lawyers (much to the horror of some of our technical Google friends) contacted manifold.net to complain that Google Earth imagery was being shown on the Manifold web site, including within this topic in the web version of Help. Google complained such images showed no Google watermarks or copyrights. Google commended our support for KML but objected to illustrations showing Google imagery. Bizarre, but true.

Because this user manual automatically gets published to the web, the only way to assure that Google imagery does not get published is not to use Google imagery in the screenshots. We have therefore replaced the background imagery from Google Earth in this topic with imagery from Microsoft Virtual Earth. It is a crazy world when Google lawyers convince software developers to feature Microsoft products instead of Google products, but that's the way it is.

As it turns out, Microsoft Virtual Earth actually works better in such applications than Google Earth, so using Virtual Earth in this topic makes most images look better.
Microsoft Virtual Earth imagery is very similar to that produced by Google Earth with several notable improvements: Microsoft Virtual Earth imagery has no “throttles” in it like Google to prevent active usage, it is not artificially disfigured with watermarks, it provides a far smoother and more realistic appearance in larger scales and in many cases it is more detailed than Google imagery. These are such significant benefits that most Manifold users have switched to using Virtual Earth for KML authoring even if the ultimate target of their authoring efforts is Google Earth.

Microsoft Virtual Earth is also very convenient because it is widely available within other geographic browsers besides Google Earth. For example, a plug in for NASA World Wind allows use of Virtual Earth in World Wind. Since KML/KMZ format is now used in very many applications besides Google Earth, it could be that a consequence of Google’s bizarre lawyering is that Microsoft’s Virtual Earth ends up being the de facto standard for image registration for KML files in all applications, even Google’s.

In any event don’t let the antics of lawyers distract you from having fun creating KML files for export to Google Earth and other applications. Whatever source you choose to use as background layers when authoring KML, it’s easy to create dramatic and useful KML files with Manifold as this tutorial shows.

See Also

A Flashy Demo - Web Queries and KML
Fun with Google Earth
Linked Images from Google Servers
Export Drawing - KML, KMZ
Export Image - KML, KMZ
Import Drawing - KML, KMZ
Other Examples
In addition to the examples set forth in the Examples topics, there are additional small examples within various topics throughout this documentation.

- **Color a Map**: Color a drawing using the Color dialog so that no two adjacent areas have the same color.
- **Selection in Drawings**: Using selection modes to select areas, lines or points.
  - **Selection Filter in Tables**: Use the Selection Filter command to show only the selection in tables while a selection is made in the table’s drawing.
- **Formatting Drawings**: Format a drawing using the format toolbar.
  - **Use Thematic Formatting to Show Territories**: Apply thematic formatting to show territories in a map that are named with text strings such as “Great Lakes” and “West Coast.”
  - **Thematic Formatting Example**: Change the size, color and style of points based on the contents of a data field.
- **Copy and Paste**: A geographic example using copy and paste within drawings.
- **Creating Areas**: Adding new areas to a drawing by drawing free hand.
  - **Autocomplete with ALT**: Create new areas or lines that exactly match the edges of existing areas or lines.
- **Editing using Smart Mouse Moves**: Move and reshape areas in drawings using smart mouse moves.
  - **How to Print**: A step by step example showing how to print using a layout, including saving a layout to a PDF file.
  - **Creating a multipage layout**: A step by step example showing the creation of a multipage layout suitable for use with map books, including an example showing how to create marginal notes citing the page numbers of adjacent sheets.
  - **Hyperlinks**: Use hyperlinks to show Congressional web pages by double clicking in a drawing.
- **Editing with Snap**: Reshape areas and move points to exact positions using Snap.
- **Reposition with Snap**: Reposition labels to the intersections of grid lines using Snap to Grid.
- **Tracing Example**: Small example drawing lines over an image.
- **Freehand Tracing**: Create an area object in the shape of Africa in a drawing layer using an image layer as a guide.
- **Create a 20 km Buffer**: Use the transform toolbar to create a border buffer zone that extends 20km.
- **Spatial Overlays and Voronoi Cells**: Given many sampling stations and a few data collection points, assign each sampling station to the nearest data collection point.
- **Transfer Values from Points to Areas that Contain those Points**: Use spatial overlays to transfer values from points to areas that contain them. A typical usage would be to sum up populations in cities within various provinces to get the total population for each province.
- **Show a 3D label in a terrain**: Show labels as 3D objects that float over a terrain.
<table>
<thead>
<tr>
<th>Examples</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Using the Layers Pane with Images</td>
<td>How the Layers pane controls work with images.</td>
</tr>
<tr>
<td>Create a Region of Invisible Pixels in an Image</td>
<td>Overlay two images in a map and cut out a region of invisible pixels in one of the images.</td>
</tr>
<tr>
<td>Create a Region of Different Color</td>
<td>Use a differently colored copy of an image together with the original image in a map and then use pixel transparency to create a zone of differing color in the combined image.</td>
</tr>
<tr>
<td>Combining Channels into Images</td>
<td>Take three Landsat 7 band grayscale images created by the Enhanced Thematic Mapper instrument and combine them into a single false-color RGB image.</td>
</tr>
<tr>
<td>Composing Complex Images in Layers</td>
<td>Create an image with drop shadows and the word “Manifold” emblazoned on it using multiple layers.</td>
</tr>
<tr>
<td>Extract scanned images into layers</td>
<td>Use the Quantize command to force scanned images into a limited number of colors and then put each color into a different layer.</td>
</tr>
<tr>
<td>Gaussian Blur</td>
<td>Create drop shadows or halos.</td>
</tr>
<tr>
<td>Georegistering an Image to Known Coordinates</td>
<td>Use GPS coordinates to georegister an image.</td>
</tr>
<tr>
<td>Create a drop shadow in drawings</td>
<td>Use the Register context menu command in a map to create drop shadow and other special effects.</td>
</tr>
<tr>
<td>Modifying Selections</td>
<td>Create buffer zones in images using the Modify Selections commands.</td>
</tr>
<tr>
<td>Painting within Channels</td>
<td>Paint into images using channels to control what colors or alpha transparency is painted.</td>
</tr>
<tr>
<td>Create Contours</td>
<td>Create contours from a surface.</td>
</tr>
<tr>
<td>Use Terrain Elevation to Enhance an Image</td>
<td>Match a surface-derived shaded elevation image to an image and then load the surface shading into the R, G and B channels of the image to create an image with enhanced 3D surface effects.</td>
</tr>
<tr>
<td>Overlay Drawings and Surfaces onto Terrains</td>
<td>Use the Terrain - Overlay command to “drape” drawings and surfaces (images, too) onto terrain surfaces in the Overlays topic.</td>
</tr>
<tr>
<td>Selections and Editing</td>
<td>Paint lines and shapes into an image and use effects where only the selected pixels respond to the painting tools or effect.</td>
</tr>
<tr>
<td>Separating Images by Channels</td>
<td>Separate an image into three grayscale images, one for each R, G and B channel.</td>
</tr>
<tr>
<td>Threshold</td>
<td>Pick out certain features from a scanned image of a map.</td>
</tr>
<tr>
<td>Threshold Color</td>
<td>Pick out certain features from a scanned image of a map.</td>
</tr>
<tr>
<td>Tracing Tools</td>
<td>Use tracing tools to create areas, lines and points in a drawing using the image of a scanned paper map.</td>
</tr>
<tr>
<td>Linked Views</td>
<td>Use the World pane to pan the view in one window based on the current view in another window. Includes navigation in terrains using a surface window as a “minimap” to guide position in the terrain.</td>
</tr>
<tr>
<td>Extract Horizontal Lines from an Image</td>
<td>Use Motion Blur and Threshold to extract horizontal lines from an image.</td>
</tr>
<tr>
<td>Importing a Projected Shapefile and Setting Coordinate</td>
<td>Importing a projected shapefile and setting coordinate.</td>
</tr>
<tr>
<td>Topic</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Shapefile</td>
<td>properties manually.</td>
</tr>
<tr>
<td><strong>Import a DXF File</strong></td>
<td>Importing projected and unprojected DXF files.</td>
</tr>
<tr>
<td><strong>Import a GML File</strong></td>
<td>How to import a UK Ordnance Survey MasterMap file.</td>
</tr>
<tr>
<td><strong>Compare Counties and Congressional Districts</strong></td>
<td>Use layer opacity to compare the shapes of counties and Congressional districts.</td>
</tr>
<tr>
<td><strong>Add an Active Column</strong></td>
<td>Create an active column that reports a discounted price using unit price and discount fields.</td>
</tr>
<tr>
<td><strong>ViewBots</strong></td>
<td>Analyze tables and make selections using ViewBots.</td>
</tr>
<tr>
<td></td>
<td>Show the sum of population for selected regions in a drawing.</td>
</tr>
<tr>
<td><strong>Create a Relation between Two Tables</strong></td>
<td>Step by step example of creating a relation between two tables.</td>
</tr>
<tr>
<td><strong>Attaching External Tables to Drawings</strong></td>
<td>How to form a relation between a drawing’s table and an external table so the drawing may use fields from the external table.</td>
</tr>
<tr>
<td><strong>Create a Query</strong></td>
<td>Creating a simple query using the <em>Nwind</em> sample database.</td>
</tr>
<tr>
<td><strong>Calculations Examples in Queries</strong></td>
<td>A series of small examples showing calculations within queries.</td>
</tr>
<tr>
<td><strong>Action Queries</strong></td>
<td>Short queries that change records throughout a table or create or delete tables.</td>
</tr>
<tr>
<td><strong>Parameter Queries</strong></td>
<td>A query that allows user input on execution.</td>
</tr>
<tr>
<td><strong>Create Linked Images and Surfaces</strong></td>
<td>Use the virtual tables of images or surfaces with queries to create linked images and surfaces.</td>
</tr>
<tr>
<td><strong>Create a Drop Shadow using the Register Dialog</strong></td>
<td>Use the Register dialog to move layers relative to each other within a map to create a drop shadow or 3D edge effect for drawing layers.</td>
</tr>
<tr>
<td><strong>Create a drop shadow in a geographic map</strong></td>
<td>The last step in an example topic mainly intended to show how to cookie cutter images using Transfer Selection, this detailed final step shows how to create a drop shadow within a geographic map.</td>
</tr>
<tr>
<td><strong>Use Convex Hull and Transfer Rules</strong></td>
<td>Use transfer rules to sum up the value of a population field in a group of points and to transfer that value into a newly created convex hull area.</td>
</tr>
<tr>
<td><strong>Find the Shortest Path between Two Points</strong></td>
<td>Given two points in a network of lines, find the shortest path between them.</td>
</tr>
<tr>
<td><strong>Match Table to ZIP Code Drawing</strong></td>
<td>Spatially geocode a table with Match to assign latitude and longitude values based on the ZIP code value in a Customers table and the location of ZIP code areas in a drawing.</td>
</tr>
<tr>
<td><strong>Regular Expressions</strong></td>
<td>Examples of regular expressions that may be used to find various patterns in text.</td>
</tr>
<tr>
<td><strong>Forms and Scripts</strong></td>
<td>Create a form and associated script that steps through the Employees table in the <em>Nwind</em>.mdb database and shows the contents of different records.</td>
</tr>
<tr>
<td><strong>Embossed or Raised Boundary Lines in Maps</strong></td>
<td>See the very end of the topic for short notes and illustrations showing an embossed effect or raised border effect.</td>
</tr>
</tbody>
</table>
About the Sample Images

This documentation uses several sample images as examples in various topics.

The bronze sample image shows the equestrian statue of Henry IV (1553 - 1610) that stands at the Place du Pont Neuf just above the Square du Vert-Galant in the romantic heart of Paris.

Henry of Navarre became the first Bourbon king of France in 1589 at the age of thirty-five when his predecessor, Henry III, was assassinated. After a life of great difficulties as well as immense accomplishments Henry IV was assassinated on Friday, 14 May 1610. He was killed by a demented man who stabbed the king twice after leaping onto the royal coach when it was slowed by traffic congestion. (The king had previously escaped another assassination attempt in 1594 when a law student had attempted to stab him at the Hotel de Schomberg.)

Henry IV completed the Pont Neuf, which was begun during the reign of Henry III. He had become king under trying circumstances without the acceptance of the population but during his reign had won the hearts and minds of the French people. Henry IV proved to be a master of practical politics and directed the expansion of French power in Europe and overseas in North America. His support for exploration yielded discoveries such as Lake Champlain. At home, the king supported arts, learning and architecture. He died a beloved king, especially among common folk. Historians have long regarded him as one of the greatest of French kings.

The first equestrian monument to Henry IV was placed at the Pont Neuf by his widow, Marie de Medici. It was smashed by a revolutionary mob in 1792. In 1818 the next generation wished once more to remember the great king and emplaced the current monument. The Henry IV we see today was poured using bronze taken from the statue of Napoleon that was removed from the Vendome column.

The schloss image shows the famous Neuschwanstein castle built by Bavaria's King Ludwig II (1845 - 1886) between 1869 and 1886. "Schloss" is a German word used for castle or chateau. Ludwig II was a man obsessed with romantic images of a medieval Germany that never was. He squandered his fortune building elaborate and expensive fairy-tale castles. The unfinished Neuschwanstein was the last and most dramatic of all.
Set where the Bavarian plain meets soaring Alps, the castle perches within a dramatic landscape of vertical crags and plummeting gorges. The photograph was shot from the Marienbrucke footbridge that vaults over a waterfall in the mountains above Neuschwanstein.

Although it is built in the imagined style of castles of medieval times, Neuschwanstein was created in the late 1800's as a dream castle for Ludwig II's personal fantasies. It is not a practical fortification even though it is sited at highly defensible location once occupied by a true, fighting castle. Numerous design themes within the interior recall epic sagas and themes from Wagner's operas. The interior is only partly completed with sumptuous rooms for the king's apartments (including an artificial cave), a Singer's Hall for musical performances and a breath-taking throne room.

The king stayed at Neuschwanstein for but a few days during the latter stages of construction. In 1886 he was beset by creditors and facing bankruptcy when a Bavarian government commission declared him mentally deranged. A delegation sent to Neuschwanstein convinced him to leave the castle on 12 June and return to Munich for treatment. The next day, the king and his personal physician, Dr. Gudden, both were found dead in the Starnberg See, a large lake on the route from Neuschwanstein to Munich. An inquiry failed to reveal whether the deaths were an accident, suicide or murder.

The Free State of Bavaria now owns Neuschwanstein and operates it as one of Bavaria's most popular tourist attractions. In life a distant and lonely man, in death Ludwig II has become a Bavarian pop icon.

The globe image shows the elevations of the entire Earth using a false color palette to code height. The data used originates in the Global Land One-km Base Elevation ("GLOBE") project at NOAA. The GLOBE project created a one kilometer gridded digital elevation database for the entire Earth.

Several hundred megabytes of data were downloaded from the GLOBE web pages to build a worldwide digital elevation database. This data was then interpolated into a lower resolution data set suitable for casual browsing and background map image generation. This lower resolution data set was then imaged using RasterLab, an internal development tool that was created at manifold.net and used for development and verification of the raster algorithms employed within Manifold System.

The height palette used in the sample image provides relatively low contrast. This image was deliberately selected for use as a sample so that the effects of various image-enhancing commands could easily be seen.
The SanFran image is a sample Landsat 7 image downloaded from a USGS web site as a .jpg image. This is path 44, row 34 using Band 3, 2, and 1. Composition of Landsat bands into a .jpg format image was done by USGS. Date of acquisition is not known but is believed to be spring of 1999. The image shows the spring runoff from the Sierra Nevada mountains and other neighboring mountains into San Francisco Bay and out into the Pacific.

Clearly visible are the numerous regions and famous landmarks of this fascinating area. In the upper right is the Delta, the confluence of the Sacramento and the San Joaquin rivers. The waters of this region drain through Suisun Bay, through Carquinez Strait and into San Pablo Bay, the large and shallow bay at the northern extremity of San Francisco Bay. The South Bay is the local term for the large, mostly shallow part of the bay that reaches from San Francisco to the South and East. A less enclosed bay is the upper portion of Monterey Bay in the Pacific Ocean, visible at the bottom of the image.

This region is the home of the microprocessor, invented by Ted Huff at Intel in Santa Clara in 1971. The Santa Clara Valley ("Silicon Valley" or simply "The Valley" to locals) is the heavily developed region at the Southeast extremity of the South Bay. The first commercial microprocessor was the Intel 4004, which contained about 2300 transistors on a die ("chip") slightly larger than one inch square.

Although well known for silicon technology, the region is also the epicenter of the venture capital funding revolution that has powered business and technology breakthroughs in software, systems, biotechnology and most recently, the Internet. It is also a center of weapons technology. The Livermore Valley appears as a horizontal streak of development just East of the South Bay. Livermore is the home of the Lawrence Livermore National Laboratory, the nuclear weapons design center created as an alternative to Los Alamos in the 1950's with an initial mission focussing on the development of thermonuclear weapons. The nuclear weapons storage depot near Suisun Bay in Concord is said to host one of the highest concentrations of nuclear weapons in the world.

This image reveals an interesting juxtaposition. On the original image (provided on the Manifold CD) we can see beige and duff yellow colored regions on the edge of the bay near Silicon Valley. These are salt ponds where seawater is evaporated in the commercial production of salt. The same region where people invent and manufacture the most advanced technology ever conceived is also the place where one of the most ancient production processes continues to be practiced.
The Manifold System splash screens used for 5.xx and 6.xx series releases show the church of San Giorgio on the island of San Giorgio in Venice seen in a view across the Canale de San Marco from the campanile (bell tower) in St. Mark's plaza. The island of San Giorgio has been the site of a monastery since 982.

The patron saint of Venice is St. George. The church in his name was designed by Andrea Palladio and was begun in 1566. Other surviving buildings in Venice by Palladio include the Basilica, the Loggia Bernardo, the Palazzo Chiericati and the Villa Capra. To get to San Giorgio from St. Mark's square, take water taxis (vaporetti) 52 or 82.

The image is taken from a photograph shot the day after the total eclipse of the sun in central Europe on August 11, 1999, which was witnessed by the manifold.net team in commemoration of the official launch of Release 5.x development.

The August 11 eclipse path crossed most of Europe's most populated areas but was cursed with cloudy weather. Manifold 4.50 was used on laptops with GPS devices to find a location between Munich and Salzburg on the totality path that (given satellite weather photos) seemed likely to be in a break between clouds. Moving every few minutes from place to place on the autobahn to stay in a clear area, the Manifold team used the atomic-clock accuracy of GPS coupled with moving-map Manifold location to find a clear spot at the exact time of totality. A fabulously clear break showed the eclipse in full, cosmic glory. Within minutes, as everyone babbled over what they had just witnessed, the clouds closed in and the rain began. The team drove to Venice over the Brenner pass and the next day shot the splash screen photograph.

In the splash screen the island has been set in a landscape created with RasterLab, the Manifold internal raster algorithm development tool used to verify terrain window algorithms for Manifold System. RasterLab allowed development and verification of raster techniques while Release 5.00 was being created. The splash screen image was composed using over 40 layers, as seen in the Layers topic.

**Historical Notes**

A few notes of historical interest, mainly thumbnail biographies, have been scattered throughout the documentation in the following topics.

George Boole  
**Expressions**

Christopher Columbus  
**Projections Tutorial**

Charles-Eugene Delaunay  
**Transform - Triangulation**

Boris Nikolaevich Delone  
**Transform - Triangulation**

J.P.G. Lejeune Dirichlet  
**Transform - Voronoi Operators**

Leonhard Euler  
**About Networks**

Johan Heinrich Lambert  
**Lambert Azimuthal Equal Area**

Pierre Simon Laplace  
**Transform - Laplace**

Gerardus Mercator  
**Mercator**
Gaspar de Portola  Example: Storing an Image in SQL Server 2008
John Parr Snyder  Space Oblique Mercator
Georgi. F. Voronoi  Transform - Voronoi Operators
Troubleshooting

Most problems reported to Technical Support arise from user errors. Almost always the answer can be found in this user documentation or on the manifold.net web site. Frequently Asked Questions (FAQ) pages. The Troubleshooting part of this manual covers various elementary issues involved in installing and using Manifold System.

Fast troubleshooting:

- Make sure your system is correctly installed. See the Read Me First topic.
- Use the Contents tab in the Help system to see if your problem is covered in the Troubleshooting book topics.
- Use the Index tab in the Help system to find topics involved in your problem. Try different key words.
- Read the troubleshooting topics cited below.
- See the Technical Support topic for information on technical support and other free resources are available.

If any error messages appeared during installation your system is not completely installed. Track down the problem and eliminate it so that you can install the system without error messages before attempting to use it. The www.manifold.net web site will have detailed installation notes in the support pages to help resolve installation problems.

Very Important: Your system is not fully installed until you have installed any service pack that has been issued for Manifold System. Do not waste time attempting to troubleshoot a particular problem until you have checked the manifold.net web site and have downloaded and installed the latest service pack for Manifold System. Because we seek a perfect release, manifold.net tends to issue service packs more frequently than other vendors.

If you can reproduce a system lockup or crash using Manifold, please send a note to tech@manifold.net with step by step instructions on how to reproduce the problem. Reproducible system hangs or crashes go to the top of our engineering priorities for immediate repair. We have found this policy rapidly results in a virtually bulletproof release.

Temporary Files and the TEMP Folder

Manifold uses the TEMP folder defined in Windows to create temporary files. The precise location of the TEMP folder is shown in the Environment Variables dialog accessible from Control Panel - System - Advanced - Environment Variables (be sure to check both user variables and system variables and be sure to check both TEMP and TMP).

By default, TEMP points to

C:\Documents and Settings\User\Local Settings\Temp

where User is the name of the user. Temporary files created by Manifold are typically named bdhXXXXX.tmp where XXXXX is any combination of numbers and letters.

If your system crashes for any reason, such as a power failure, while using Manifold there may be one or more large temporary files left in the TEMP folder. These may safely be deleted if Manifold is not running.

See Also

See the following topics for problems with....

Installation or Configuration

Problems when using Windows 95, 98 or Me
Importing or Exporting
Commands or Tools
Tables
Performance

Projections

Internet Map Server
Problems with Installation or Configuration

If you are reading this Help file, obviously enough of Manifold System has been installed so that Manifold can be launched and this Help file run. This simplifies troubleshooting.

Error Messages During Installation

- You must have Administrator privileges when installing. If using XP Home, you must login with the true Administrator account since some other accounts with "administrator" privileges do not have full administrator privileges.
- You must have a reasonably recent Microsoft Internet Explorer browser installed on the system.
- No other program should be running during installation. This includes any programs (except Microsoft Windows system processes) running in background that get launched automatically on startup. Disable all virus checkers, network snoopers, "value added" uninstall utilities and other programs.
- In Windows '98 and Me it is possible that a previously installed program broke your Windows dll's by replacing them with out-of-spec dlls or that the program failed to update your Windows dll's in the correct way. If you have installed a lot of shareware or programs from non-Windows vendors this might have happened. The fastest way to resolve this is to wipe the hard disk clean (reformat it completely) and then re-install Windows. Re-installing Windows without re-formatting the hard disk may not solve the problem since the re-installation may attempt to "preserve" out-of-spec dlls or registry entries. This is less of a problem with Windows 2000, XP or Server 2003.

Manifold won't launch. Menu items not available. Error messages on some menu choices. Can't execute scripts.

- Are you using a Runtime serial number? Runtime serial numbers do not allow interactive use of Manifold System.
- Were there any error messages of any kind during installation? If so, the installation is not complete and Manifold may not be fully functional. Resolve the installation difficulty.
- If the system was functional and is no longer functional in some way, has anyone installed new software that altered or deleted Windows system files on which Manifold relies?
- Has any user customized the system with add-ins or other customizations that remove menu items?
- Most toolbars are on by default and can be turned on/off via Tools - Customize. Best efforts have been made in this manual to show only default configurations; however, it is always possible that some illustrations show toolbars that must be enabled via Tools - Customize. Check to see if another user has not changed your configuration.
- Manifold menus and toolbars are context sensitive and will change depending on the type of component that is active. If a drawing window or layer is active, menus and toolbars will show commands and tools that apply to drawings. Image commands and tools will appear only when an image window or layer is active.

Toolbar settings or other options have changed unexpectedly.

- Has anyone else used your user account and changed the settings?
- Are you using a different Windows user account? All of Manifold installs as a shared application so that all user accounts may use the system. All option settings such as toolbar and window positions and any customization settings are personal to the user account that set them up. If you customize the system using one Windows user account and then login and launch the system using a different Windows user account you will not see the customized settings.

Transform toolbar is missing operators.

- Check which window or map layer has the focus. The transform toolbar will be loaded with operators that apply to the component type that is active. Click on the window or layer you want to work with and the toolbar will be configured with the operators that apply to that window. Operators for tables, images, and drawings are different because the data these components contain is different.
- Check the data type for the fields in your database. Remember that what appear to be numbers, like "3.56" can in fact be text string types and not numeric field types. You may think you have a table full of numeric fields when the table might have only string fields. In that case, the transform toolbar will offer only operators that may be applied to string fields and not to numeric fields.
No panes are visible. Project Pane has disappeared.

- Panes are not displayed until they are checked in the View - Panes menu.
- Toggle panes on/off with SHIFT-ALT-(letter) combinations. SHIFT-ALT-P turns the project pane one and off.

No scroll bars are visible.

- Manifold does not use scroll bars for images, drawings or maps. Hot scrolling using CTRL-right click is used instead because it is faster and uses less space. See the Windows topic.

Windows disappear when minimized

- If the project pane or other pane or window has disappeared after being minimized, check the vertical scroll bar on the right side of the main Manifold window. Windows can be positioned within Manifold so that they are partially out of the main window. In that case, a scroll bar appears. It could be that when the missing window was minimized it appears scrolled out of site in the main Manifold window.

Display windows refresh and then show blank contents

- In Tools - Options - Miscellaneous, check the Use GDI-compatible offscreen cache option. The problem is caused by a buggy video card driver (are you using an ATI card?) and this option uses a lower performance workaround that avoids triggering the bug.

Miscellaneous

- For normal operation you should have four times the maximum size of the files in use available in free disk space. If you are working with a 100 MB image you should have 300 MB free on disk. The extra space is needed to keep temporary versions of the image to allow Undo or abandoning of edits as well as to provide space for copy and paste operations.
- See the Memory Requirements topic for additional information on RAM and disk space requirements when working with larger projects.

Disk Space

- Always double-check you have enough free space on hard disk. If you have no free space on hard disk, Manifold will not be able to function because there will be no room for temporary files.
- Even if you have plenty of free disk space, check the location of the TEMP folder for Windows. If the TEMP folder is placed on a disk partition that does not have enough free space, Manifold will not be able to function correctly.
- If you work with linked images, clean out your cache files from time to time. Cache files persist forever and, in extreme cases, use up your free disk space unless they are manually deleted. See the recommended actions in the Managing Cache Files topic.
Problems when using Windows 95, 98 or Me

If you are using a Windows '95, Windows '98 or Windows Me operating system, before troubleshooting any problems with Manifold please read the Limitations topic. Specific problems covered in this topic include:

- Areas not always displayed in Windows '95.
- Zoom limitations in Windows '95.
- Crashes in Windows '95 and Windows '98.
- Printing limitations in Windows '95, '98 and Me

This topic has been placed here to catch anyone browsing the index and to redirect him or her to the Limitations topic.

See the Memory Requirements topic for additional information on RAM and disk space requirements when working with larger projects in Windows 95, 98 or Me operating systems.
Problems Importing or Exporting

Problems importing or exporting to different formats account for most Technical Support issues. Most such problems are caused by confusion arising from the limitations of legacy formats, especially when importing projected maps. The most confusing situation on import arises when a legacy format unsuited for the saving of projected data (such as .dxf or .shp) is used to save maps projected in an intrinsically user-unfriendly projection such as UTM.

On export, it is important to remember that legacy GIS systems such as ArcView and MapInfo are not able to handle all aspects of rich data that may exist within a Manifold project. Any export will inevitably involve manual intervention to simplify the data into those limited forms that the target system can handle. If you are exporting data for use within legacy systems it is your obligation to learn enough about the limitations of those systems to know how data must be dumbed down into something they can handle.

Importers do not work correctly or report errors.

- Are you sure you are using the correct importer? Several GIS formats use the same file extension but require different formats. For example, both NIMA NITF and UK Ordnance Survey NTF formats use a * .ntf extension for files. Files may be described casually and inaccurately as something other than what they are. The generic term “DEM” refers to at least five different formats, each of which requires a different importer.
- Is the file the format you think it is? Errors occur on web sites that provide files. The USGS web site, for example, has on occasion incorrectly provided FTP access to DEM files that were really SDTS DEM files and vice versa. Directories providing access to “DLG” files will often provide them in SDTS form.
- Is the file damaged? When working with a downloaded file, try downloading the file again in case an error was introduced during download. On rare occasion files will be damaged when loaded across a local network or when read from a damaged hard disk sector.
- Has the file been decompressed? Files are often published as compressed archives using “zip” or “GNUzip” or “tar” archival or compression methods. These must be decompressed or unzipped before use with Manifold importers.
- Has the file been correctly decompressed? When decompressing UNIX tar or tar.gz files with WinZip one must uncheck “TAR file smart CR/LF conversion” in the configuration settings to avoid introducing errors during the decompression process. Other decompression utilities may have similar nuances involved in correct operation.
- Do you have adequate free space on your hard disk and on the hard disk volume hosting your TEMP file? You should have free space that is three times the size of the imported file.
- Have you selected the right importer? Some GIS data files do not have extensions and so the right importer must be selected.
- When importing DLG files, are you sure they are in “optional” DLG format? This is the format imported by Manifold and has replaced the “original” format. DLG optional replaced the original DLG in government usage many years ago. However, on very rare occasions one may still encounter DLG files in “original” format.
- Files are sometimes provided without extensions when they should have extensions. For example, the 14 CD set of DLG files covering the US that was published by USGS in 1993 contains the DLGs in the form of zipped files. However, when unzipped, the result is files that are named using numbers without a three letter extension, for example SF4RDF04. These must be renamed to add a .opt extension to each file, for example SF4RDF04.opt.

Imported drawings appear to be incorrect.

- Is the imported data projected? Projected maps saved in old formats will require manual entry of the required projection formats. See Projections and Legacy Formats.
- Does the imported drawing represent a geographic drawing? If it is a CAD drawing it will require georegistration to appear in the correct location and scale.
- If importing a CAD drawing for non-geographic use and the scale is incorrect, read the topic for the importer being used to learn how to set scale when importing.
- Manifold importers use reasonable default settings for formats but individual drawings may require choices other than default settings depending on the specific nature of the file being imported. Read the Help topic for the importer being used, read any documentation that might accompany the file being imported and try again with different settings.
- If an imported projected geographic drawing does not match other drawings it may have been published using a different datum. Re-project it using the same datum as the drawings it is to match.
- Some drawings may be the victim of some fundamentally bad choice made by the publisher that irretrievably damaged their accuracy. Data on the edges of UTM zones in maps published in UTM
Troubleshooting

projection might suffer from this problem. Try to get the data in some other form, preferably unprojected.

- Some formats very poorly organize objects in drawing. Certain SDTS files, for example, can place large areas in the same drawing where they overlay and conceal smaller areas. Until the larger areas are deleted or moved to a different drawing the areas underneath might not be visible.

Imported drawings do not look like original maps.

- Are you sure you are comparing the same thing? GIS systems often have an option to save in different formats. The information saved by the exchange format you are using to import a map might not contain the same formatting information (colors, etc.) as that used by the GIS system to display a highly-formatted map.
- Formatting applied to the underlying objects drives the appearance of drawings or maps in the original system. This will be different from system to system. Not all exchange formats will convey formatting information in a way that’s usable by Manifold. See the Formatting topic for general information and the specific topic for the importer being used.
- If text or other embellishments do not appear it could be that the GIS system from which you are importing maintains such capabilities in a way that is not translatable into Manifold. See the Help topic for the importer being used.

Imported maps do not look like pretty road maps.

-Maps imported from most generic sources will require formatting. Read the Key Ideas topic and the Formatting topic.

Imported drawings or images seem empty, do not appear in maps, or appear in unexpected places.

- If the drawing or image does not appear in the map, use Zoom to Fit to make sure all of the data is in view. Select the image or drawing and GoTo the selection and zoom far, far in to make sure you don’t overlook anything. It could be that the item is visible in the map but is not in the location expected (and so is far away from the view seen in the window) or is so small (because it has not yet been scaled properly) that it is not obviously visible.
- Did you import from a geographically aware format? Import from non-geographic formats such as .dxf or .jpg will require georegistration to correctly locate the drawing or image.
- Did you import a projected drawing or image? If importing from a "dumb" format you may need to manually specify projection information. See the Projections topic.
- If the imported items appear in a tiny dot off the coast of Africa you have imported from a non-geographic format and must georegister the image or drawing or must manually specify projection information.
- Are you sure the drawing or image contains the data you think it does? Open the drawing or image within a drawing or image window to see how it looks in its native coordinates. For drawings that appear empty, use Zoom to Fit and change the formatting of points to a large size so that any points will not be overlooked. Drawings that you expected to contain lines and regions might contain only a few points that are so few and so small that the drawing appears empty. Open the drawing’s table to see if there are any records.
- If images appear empty it is possible (but very rare) that they are RGBA images with very high transparency set on each pixel. The pixels are there but are so transparent they cannot be seen. Check to see if the image is an RGBA image. If it is, click OFF the Alpha channel in the Layers pane so you can see the image without any transparency effects.

Problems importing or exporting to ESRI .shp "Shapefiles".

- Are you having trouble importing a shapefile that might contain projected data? A recent hack to the shapefile format incorporates projection information but most shapefiles that have been published predate this extension and so do not save projection information. If you use such shapefiles to import projected data you must manually add the required projection information. See the Projections and Legacy Formats topic.

Problems importing geocoded data from Excel.

- It’s not a good idea to keep databases or database-style lists of records in Excel because Excel does not have the built-in discipline to keep databases neatly organized that is provided by database
management programs such as Access. It's easy to go wrong in Excel by creating spreadsheets that are not really database tables, that co-mingle column headings and comments with record information, or that specify unexpected field types. The Manifold geocoded data importer cannot repair a confused data layout in Excel. The best approach to debugging a "database" stored in Excel is to first import it into Access. If you cannot import it into Access it is not a valid database table and Manifold will not be able to import it either. Organize the Excel file so it can be imported into Access. Better still, use Access in the first place.

Problems importing geocoded data.

- Is the problem reading latitude/longitude coordinates? Manifold reads all of the common ways of writing latitude and longitude values, but some data sets use such weird formats for latitudes and longitudes that the Manifold importers cannot be adjusted to read them. In such cases the latitude/longitude values must be translated into a standard form. We recommend standard decimal notation where each latitude or longitude consists of a whole degree value followed by a decimal point and decimal fractions of a degree: "38.0852". Latitudes from the Equator to the South pole are prefixed with a minus sign: "-39.0852", as are Longitudes from the Prime Meridian westward to -180 degrees longitude. It's possible to use Manifold to make conversions from really strange formats if you know how to use the transform toolbar. To make a conversion, import the database as a table and then use the transform toolbar operators to copy/translate the latitudes and longitudes into standard form. Token operators will be used, so learn how to use them. You can then export the table and then import it as a geocoded database to create a drawing showing the information it contains as points.

- Is the problem reading coordinates that are supposed to represent projected data? Read the Projections and Imported Components topic and the Projections book topics. Importing geocoded databases where the coordinates are for some projected coordinate system (such as UTM or Gauss Kruger) is expert-level stuff that involves a host of complications. You'll need to import the geocoded file as abstract coordinates and then manually set the coordinate properties to the exact projection and projection parameters that were used for that data.

- Did you try to import a list of addresses and failed? A database that has no location information other than addresses is not geocoded. To be a geocoded database, it must have a latitude and longitude field for each record that shows exactly where that address is located. Databases that do not have a latitude and longitude field for each record must first be geocoded so these field values are computed for each record.

- Did you expect to get a map and instead got a drawing full of points? Geocoded databases should be tables full of records that include latitude and longitude fields for each record. Importing such a geocoded database will create a drawing that contains points only. Each point is the location of one record.

- Did you try to import a geocoded database that you thought represented lines and areas coded as a series of records and failed? There is no GIS standard for representing lines or regions in the form of geocoded databases. You may have run across some data that purports to show map features in the form of ASCII lists of records or some other tabular format; however, if it is not one of the standard formats it cannot be read by Manifold. Use one of the formats supported by Manifold to import drawings that contain objects other than points.

Imported Tables have wrong field types or missing fields

- Different database formats save data using different approaches to field types. Some primitive format such as .txt or .csv don't save field type information at all. In such cases the importer must guess what data type should be used for that field. It is possible the importer will guess wrong. For example, it might import what you think is date time information as a text string. Use the Transform toolbar to copy/translate the data into the desired field types.

- When importing from complex formats such as SDTS a file may be imported as many different drawings. Some of them may have certain data fields while others do not. Look through all of the drawings created when looking for given data attributes. Also, some of the more complex formats (like SDTS) may require checking optional import boxes to map fields between tables and drawings.

Exported .mdb databases cannot be read by Access 97


Problems importing or exporting to .dbf format
• Have you made sure your Windows system is current as described in the installation instructions? Manifold uses Windows drivers to read and write .dbf files. If your Windows system has not been correctly updated to incorporate the latest Microsoft bug fixes and other current components you may not have proper Windows functioning in this important area.

• Some programs (especially those in UNIX) do not write .dbf files that are in accordance with Microsoft standards. The fastest way to resolve most problems is to make sure that all names (including file names, field names and so on) have no more than eight, plain alpha characters. See the Importing Tables topic.

Cannot write when exporting

• Are you sure you are writing to a directory for which you have write permissions? Your user account may not be authorized to write into that directory. Try writing to a different location.

• Are you trying to write onto a CD ROM? CDs are not writable. Try writing to a location on hard disk.

• Are you trying to overwrite a read-only file? If you copied a file from DVD ROM, in Windows that file will continue to be read-only even though you copied it onto the hard disk. The file had "read only" properties set when it was on DVD and copying a file in Windows copies all of the file's properties as well. You need to right-click on the file in Windows explorer and change the properties to make it writable.

• If you are trying to write into an external database you may not have correct permissions to write into that database. Check with your database administrator to see if the database is writable or if any special passwords or other access methods are required.

• Are you trying to write information that is not supported by the format? For example, a field name such as "My really long field name for Site $325" cannot be exported to ESRI .shp format due to the limitations on the number of character names allowed in field names in this format. Manifold's exporters will attempt to auto-truncate such names; however, a really determined user may find it possible to defeat such safety measures.

• Did you see any error messages during installation of Manifold? Manifold must install the latest versions of various Microsoft facilities for accessing data. These include ADO, Jet, DAO, ODBC and OLE DB files if your system does not currently have these standard Microsoft facilities for data access. If any other application is operating, some of these files may be locked and can not be updated. Make sure no background processes are running and re-install Manifold.

General inability to import tables

• Did you see any error messages during installation of Manifold? Manifold must install the latest versions of various Microsoft facilities for accessing data. These include ADO, Jet, DAO, ODBC and OLE DB files if your system does not currently have these standard Microsoft facilities for data access. If any other application is operating, some of these files may be locked and can not be updated. Make sure no background processes are running and re-install Manifold.

Terrain Windows are Blank

• Manifold requires a functioning OpenGL subsystem to display terrains. If there are no OpenGL capabilities in the system terrain windows will be blank when opened.

Miscellaneous

• For normal operation you should have four times the maximum size of the files in use available in free disk space. If you are working with a 100 MB image you should have 300 MB free on disk. The extra space is needed to keep temporary versions of the image to allow Undo or abandoning of edits as well as to provide space for copy and paste operations.

• See the Memory Requirements topic for additional information on RAM and disk space requirements when working with larger projects.
Problems with Commands or Tools

Manifold uses a context-sensitive, mode-oriented user interface for many command functions. Unexpected action of commands or tools is usually a result of having the focus being on a different window or being in a different command mode than was thought. Many Manifold commands and tools are designed to work within the selection or require a selection to be present; therefore, it is important to note whether a selection is present or not.

The first step in any troubleshooting is to check which window or layer has the focus. If a drawing window or layer is active the toolbars and menus will be configured for drawings. Image commands and tools will not be available unless an image window or layer is made active. See the Windows topic for information on what the “focus” means and what it means for a window to be active.

The second step in troubleshooting is to see what tool or command mode has been selected. If the Select Box button is pressed in the mouse will be in selection mode and cannot be used to zoom in or zoom out, for example.

The third step in troubleshooting is to see if there is any selection in the active component. Commands and tools will change their action to work within the selection, if one has been made. Some commands, such as Crop, work only upon the selection.

Unexpected command or tool action. Command or tool doesn't work.

- What is the active window or layer? Clicking back and forth between images and drawings will automatically switch the context of commands and toolbars to feature only those commands or tools that apply to the type of component at hand. Images and drawings are different types of components with different characteristics. Although some tools are similar and are usually directly analogous in both cases the precise action of tools will be different in drawings or images.
- What is the command mode the mouse is in? If you have chosen Zoom Box, the mouse will be in zoom box mode and not selection mode. Get in the habit of clicking out a tool button once you are done with that tool.
- Will the command or tool cause a visible difference? Some commands may not have any visually obvious effect. Perhaps the command worked correctly but had no visually obvious effect. For example, converting an RGB image to an RGBa image will not cause any difference in the appearance of the image.
- Does the command require a selection? Some commands such as a Cut or Crop work on the selection. If no selection has been made the command does nothing.
- Is the command affected by a selection? Commands such as transform toolbar operators will use autoscope, where they will apply to the entire component if there is no selection and will apply only to the selection if one has been made.
- Is there a selection? Most tools will work only within the selection. If you have made a selection and are trying to paint outside the selection there will be no effect. See the following paragraphs for cases where there might be a selection but it is not visible.
- When working with tables, check the field type. It's possible that what you think is a number is really a string field, for example. The specific action of commands and tools that work with tables will often depend on the type of the field involved.
- When working with images, check the settings in the tool properties pane and format toolbar. Painting with a size of 1 will have a greatly different effect than applying the same brush shape with a size of 20. Likewise, the Tolerance setting has a great effect on Touch Selection and the Bucket tool.
- Is the system busy executing a previous command? Check the System Activity indicator in the in the status bar at the lower right hand of the screen. When the system is busy it will show a 3D “bump”. If Manifold is busy it cannot execute a new command for you.

Objects created in Drawing in a Map do not appear.

If a map with several different drawings is open and you try to draw a new object in the drawing and it does not appear:

- Does the drawing being used have the same projection as the map? Right click on the layer tab for that drawing and choose Project to Map to force that drawing to use the same projection as the map.
- When creating drawings to be dragged into the map, make sure you click on the opened map first to activate it. While the map is active, click on File - Create - Drawing to create the drawing. If you open the map and click on something else (like, the project pane), the focus will move off the map window and the new drawing when it is created will not be created with the projection used by the map.
Troubleshooting

- Try **Zoom to Fit** to make sure all objects are visible.
- Does the drawing have 100% transparency set? This will make the drawing and anything in it total invisible because it is completely transparent.
- Is the drawing hidden from display? Double click on the layer tab to toggle it ON.
- If the objects are areas, is the drawing formatted so that the foreground color of areas is invisible color? If so, the areas will be invisible since this is a shorthand way in Manifold of turning off areas.

**Cannot change the format color of objects**

- In a map, have you clicked on the layer in which the objects are located?
- If you cannot change the color of red objects... are they selected? If so, they will be seen in red selection color no matter what their color is supposed to be.

**Items are not selected. Selection tools have no effect.**

- Is the selection visible? Check the **Selection Style** to make sure you have not turned off showing the selection. You might be making the selection OK and just cannot see it.
- In a map, have you clicked on the layer in which the objects are located?
- Is the selection very small or not in view? You might not realize there is a selection because it is only a few pixels in a large image and is not visible, or because the selected region is scrolled off screen. It might be off screen when making selections from table windows or via SQL or other non-mouse selection methods.
- Have you enabled selection for the types of objects you wish to select? If **Select Areas** has been enabled and **Select Lines** and **Select Points** have been disabled, any selection made with the mouse will apply to areas only. Making a mouse selection over lines or points will have no effect because they are not enabled for selection.
- Are you using a "closed" selection tool when an "open" tool is required? Default selection tools select objects that are entirely within the mouse selection zone. You may be trying to select a line, for example, that partially extends outside the selection zone drawn with the mouse cursor. Use **SHIFT** with the selection tool to get an "open" style of selection that will select all objects any part of which are within the zone drawn by the mouse.
- Maps will display the selection for all components/layers within them. It is possible to select areas in a drawing that is seen in a map layer above an image layer, where the selected area in the drawing appears to be a region of selected pixels in the image. This might occur in certain combinations of area styles and transparency effects for images and drawings. Click layers on and off to resolve any visual ambiguity.
- It is important to understand the different types of objects that appear in drawings. Trying to select the "inside" of a region in a drawing that is outlined by lines and is not an area object will have no effect because there is nothing there to select. See the Drawings topic for a discussion.

**Missing objects in drawings or missing regions in images**

- Have the objects or pixels been deleted?
- In drawings, have the objects been formatted to the same color as the background? Click off the **background** in the **Layers** pane.
- Missing points? Increase the point size so that small points are visible.
- Are the objects or pixels on screen? Click **Zoom to Fit** to make sure that everything within the drawing or image is within the window that's open.
- In drawings, are the objects smaller areas covered by larger areas? Click using **Touch Select** on any spot and see how many objects are selected. If two objects are selected where you think there is only one area there is another area under the area that you see. Select the smaller areas in a table window and move them to a different drawing. You can then organize them in layers in a map so they are above the larger areas.
- Missing regions in images might simply be regions of very high transparency in RGBA images.
- If you are running Windows '95, the operating system cannot display objects made up of more than 32,000 coordinates. See the Limitations topic. Switch to Windows 2000.
- If the objects are areas, is the drawing formatted so that the foreground color of areas is invisible color? If so, the areas will be invisible since this is a shorthand way in Manifold of turning off areas.

**No panes are visible.**
Panes are not displayed until they are checked on in the View - Panes menu or with a **SHIFT-ALT** (key) keyboard shortcut.

No scroll bars are visible.

- Manifold does not use scroll bars for images, drawings or maps. Hot scrolling using **CTRL-right click** is used instead because it is faster and uses less space. See the Windows topic.

Some toolbars are missing.

- The default installation of Manifold System contains the toolbars described in this documentation. If toolbars are missing or do not appear in the **Tools - Customize** dialog, check the **Config** folder within the Manifold System installation folder for an **XML** files that may have been added containing `<addin>` entries. An add-in may be used to add commands to Manifold and also to control the availability of toolbars.

Objects disappear when pasting into drawings.

- Were the objects selected? Manifold's Paste into drawings is a standard Windows Paste: if something is highlighted/selected when a paste occurs, that something gets deleted and is replaced with whatever is being pasted in from the clipboard. See the Using Cut, Copy and Paste in Drawings topic.

- Were the objects copied from a drawing that was not correctly georegistered? If you create or import a CAD drawing from a non-geographic format (such as DXF) the objects are created in Orthographic projection at the 0,0 intersection of the Prime Meridian and the Equator. If you then paste them into a correctly georegistered drawing (such as, for example, a drawing of city streets in some town in Europe or the US) the objects will not be seen in the drawing. They are in the drawing but they are still in a location somewhere off the coast of Africa at the 0,0 intersection. To remedy this, correctly georegister all drawings that will be used.

- If the objects are areas, is the drawing formatted so that the foreground color of areas is invisible color? If so, the areas will be invisible since this is a shorthand way in Manifold of turning off areas.

Compressed images don't appear in maps.

- A compressed image may or may not be displayed in a map, depending on the projection used for the image and the projection used for the map. If the projections are not the same, the compressed image will not appear in the map unless the projections are sufficiently similar to each other (not usually the case) so that the conversion between the two can be accomplished by applying an offset and altered scale.

Data attributes disappear or do not appear as expected after an operation.

- Many operations such as those in the Spatial Overlay dialog require correct setting of transfer rules for the fields that are to be transferred.

Strange Changes in object shapes when pasting into drawings
If we **Copy** selected objects like those at the left and **Paste** them into a different drawing and get the result at the right, the problem is that the drawing on the right uses Orthographic projection while the drawing at left uses Geographic Latitude Longitude projection. If a drawing is created when the project pane has the focus, by default it is created using the Orthographic projection. When the objects were pasted into the Orthographic drawing they appeared as they would be shown in that projection.

Situations like the above arise when we selected the objects in the lat-lon drawing, chose **Edit - Copy** and then clicked on the project pane and chose **File - Create - Drawing**. At the moment we clicked on the project pane we moved the focus to the project pane and so the new drawing was created in Orthographic projection. What we should have done was choose **File - Create - Drawing** while the lat-lon drawing was still the active window. In that case, the new drawing would have been created by default to match the drawing that was open and had the focus.

Note that the pasted North and South America objects are correctly seen in default Orthographic projection, which is, after all, a view from space centered on the intersection of the Equator and the Prime Meridian just off the coast of Africa. From that position North and South America appear greatly foreshortened as they are on the edge of the visible part of the Earth. The default Orthographic projection uses a "transparent globe" so that Alaska in the drawing at right is seen through the Earth. See the Projections topics for more information.

We can retrieve the situation by re-projecting the drawing on the right into Geographic Latitude Longitude projection.

**Problems printing. / **"No System Resources"** error when printing**

- Make sure you are using the latest version of Manifold and the latest Service Pack.
- What version of Windows are you using? Windows 98 and Me are full of bugs that often affect printing. Use Windows Server 2003, Windows XP or Windows 2000 to prevent bugs in Windows from interfering with print jobs.
- What printer resolution are you using? Try decreasing resolution and see if it helps. If you print in color, try black and white. If either of these steps helps, it is almost certain that the problem arises from a bug in the printer drivers supplied by your printer manufacturer. Download the latest version of your printer manufacturer's drivers to see if they remove the bug.
- How much RAM memory do you have installed in your system? 32 or 64 MB are not enough for serious printing jobs and 128 MB is only marginally OK. At least 256MB of RAM should be installed in the system.
- How much free space do you have on your hard disk? Print jobs can require lots of free space for temporary files.

**"Unknown Error" or "No System Resources" or "Can't Read Data Stream" errors when working with large images or other commands**

- 32-bit Windows editions are fragile when large processes are used. Switch to 64-bit Windows so that Windows errors do not interrupt work.
- Such errors are frequently caused by running out of free space on hard disk. Check the amount of free space on hard disk. If working with very large files and a system crash occurs (such as during a power failure) Manifold temporary files beginning with *bdh* may be found in the system TEMP folder and may be very large. These *bdh* temporary files may be safely delete after a crash to save space on disk.
- Errors may also be caused by limitations of the particular Windows configuration in use or by Windows bugs in Windows '95, '98, '98SE or Me. Check all items suggested for the Problems with Printing section above.
- Are you using Windows Server 2003, Windows XP or Windows 2000 with the latest Microsoft service packs installed? If not, the above messages are almost certainly caused by memory management or other bugs in older Windows releases.
- Make sure you have lots of free space on hard disk. Do you have a few gigabytes of free space on hard disk? If you don't have enough free space on disk to allow Manifold to create and use necessary temporary files the system cannot operate.
- Do you have a static Windows pagefile (where the minimum size is the same as the maximum size)? Has it been set to a very large size? If not, you could be running out of pagefile when Windows attempts to swap parts of a large job in and out of memory. Set up a large, static page file.
- Make sure your TEMP directory is on a hard disk that has lots (many gigabytes) of free space.
How much RAM do you have? Running with lots of RAM is a good way to avoid stepping on Windows bugs.

If some operations work OK and others do not, try saving the project to a .map file prior to the operation to move image data from memory to file mapping.

Keep in mind that images stored in compressed formats will expand to their real operational sizes when used in Manifold. Thus an image that takes only 200 MB in some compressed format could end up being a gigabyte in size when actually used. Make sure you have lots of free disk space to be able to work with such images.

Problems Saving or loading Projects

• Encountering errors of the form "Can't overwrite target file. The project is saved to [path] bdh17a.tmp"? This happens when you don't have enough space on the target disk drive or when the temp file or target file can not be read or written due to a locking error. Locking errors might occur because you have opened the temp file or the target file name in an editor, such as a hex editor, or have the target .map file name opened in another instance of Manifold. Rename the .tmp file given to a .map file to rescue the project that was saved under the .tmp file name.

• "Incorrect Data Check" errors indicate hardware problems, not Manifold problems, and usually arise from disk errors. This error message may occur seemingly at random as disk errors are encountered in an unpredictable way. Windows places information in scattered places on disk as free blocks become available when files (including temp files created or deleted by different programs or system processes) are created or deleted. Depending on the semi-random usage of blocks, a particular save or load may or may not involve a bad patch on disk. Note that disk is used in loading a project, because decompressing the project involves temp files created on disk. Likewise, saving a project to disk will involve not only the actual disk blocks used in the saved project but also disk blocks used for temp files created temporarily during the compression process. Regular use of Windows disk maintenance utilities can help avoid such problems; however, as systems age intermittent disk failures may indicate unreliable motherboard or disk circuitry and can be difficult to diagnose or eliminate.

• "[Can't overwrite target file ...]" errors usually indicate flaws in virus-checking software. The overwriting of the old file with the new file is done using a system routine, which gets called after Manifold closes both the old file and the new file. When a system routine is not able to overwrite the old file, some other process besides Manifold has locked it. This behavior is frequently experienced with virus scanners and other protection software. Trying to change the name in background won't work either: By the time you see the error message, the new file is already closed. In fact, it is closed even earlier, prior to the call that generates the error in the first place. Not being able to rename the file is yet another indication it has been locked by some other process besides Manifold. Important: Virus scanners and other protection software are frequently designed to inject a portion of their code into every running process. In this case, the behavior mentioned above can seem to be caused by the Manifold process, despite the fact that it is being caused by software external to Manifold that has been injected into the Manifold process.

If the geocoder does not work, check the following:

• Manifold Geocoding Tools extension product has been installed.
• All state files required are present (that is, required files have not been removed after installation).
• The Geocoding Database folder in the Tools - Options - File Locations correctly specifies the folder used to install the geocoding database. By default, this location is C:\Program Files\Manifold System\GCDB.
• If geocoding functions do not work within IMS, check that the geocoding database has been installed within the main Manifold installation folder, such as in the default path above.

If an address cannot be located in the geocoder, check the following:

• The address is correct. Nowhere is the ancient maxim of computing, "Garbage in, garbage out" more correct than in street address geocoding.
• The address does not contain secondary address information, such as "Apartment 20" co-mingled with the primary street address field.
• The address is a real street address and is not a virtual address, such as a Post Office box address (which, of course, can only be geocoded to a zip code).
• Try manually breaking up a single line address into street, city, state and zip fields. If the zip code cannot be found, check the zip code against the US Postal Service's website to make sure it is a valid zip code.

Miscellaneous
• For normal operation you should have **four times** the maximum size of the files in use available in free disk space. If you are working with a **100 MB** image you should have **400 MB** free on disk. The extra space is needed to keep temporary versions of the image to allow Undo or abandoning of edits as well as to provide space for copy and paste operations.

• See the Memory Requirements topic for additional information on RAM and disk space requirements when working with larger projects.

**General Advice**

Run 64-bit Windows using 64-bit Manifold. Doing so eliminates an entire class of 32-bit Windows bugs.
Problems with Tables

Most routine problems with tables arise from confusion over field types. The classic problem is working with a series of values like "3.145", "2.44" and so on and thinking they are numeric types when in fact they were imported from a database that stored them as text strings. Remember, in databases there is a difference between a numeric value that is 3.145 and a text value that has the ASCII characters "3", ".", "1", "4", and "5" arranged in a row. The numeric value is a number that can be multiplied, divided and so forth while the text string is just text characters that spell out something we read as a number, like the phrase "three point one four five."

General inability to import tables

- Did you see any error messages during installation of Manifold? Manifold must install the latest versions of various Microsoft facilities for accessing data. These include the latest MDAC as well as MDAC 2.5 if your system does not currently have these standard Microsoft facilities for data access. Make sure no background processes are running and re-install all required Windows updates as set forth in the Manifold installation instructions. If any errors occur during MDAC installation or other Windows updates, consult the Microsoft Knowledge Base to resolve the problem that is preventing your Windows system from being correctly updated.

Slow performance with SQL Server, Oracle or other OLE DB server

- When working with server-based OLE DB providers such as SQL Server and Oracle, users are strongly encouraged to maintain primary keys in all tables linked into the Manifold project. A side effect of how such servers interact through OLE DB is that if the table does not have a primary key, performance will be greatly reduced.
- If read-only access is acceptable, ADO.NET is often the fastest possible connection and should be used in preference to ODBC or OLE DB for that reason. Important: Tables linked from ADO.NET data sources are always read-only.

Trouble linking tables from an Oracle DBMS

- Always set the Allow saving password option. If this option is not set, the system will connect for the first time but not at all other times. The intricacies of the OLE DB / ODBC interaction that takes place within the Microsoft OLE DB Provider for ODBC Drivers are such that it is almost never possible for the Manifold to jump into the middle of the connection process and prompt the user for a username and password.
- Avoid using ODBC. Use the native Oracle (OCI) connection offered by Enterprise Edition. If you do not have Enterprise Edition and only need a read-only connection, use ADO.NET Oracle. If you do not have Enterprise Edition and require a read-write connection, or if you absolutely have to use ODBC, use either the native Oracle ODBC driver or a combination using the Microsoft OLE DB provider for ODBC with the Microsoft ODBC driver for Oracle, depending on what works best in your configuration. Many native Oracle ODBC drivers are limited to forward-only cursors, which is by far the simplest type of database cursor allowed in ODBC, and thus have limited functionality in Manifold.

I'm having problems with ODBC.

- General problems with ODBC: this is a standard Microsoft technology that is part of Windows. Get a good book on ODBC and Microsoft-stuff to learn more. Note that Manifold uses OLE DB to connect to ODBC. Avoid using ODBC if possible. Instead, use OLE DB.

Where do I find an ODBC driver for my database format?

- You don't need ODBC to open standard Microsoft database file types of the sort that can be opened directly with Access: dbase, FoxPro, Paradox, Excel, Lotus, text and .csv, and HTML tables. These can be opened directly using the correct selection in the Files of Type box within Manifold Open dialogs.
- If you don't already have them, the installation of MDAC 2.7 and 2.5 as set forth in the Manifold installation instructions will install ODBC drivers for the most popular database types (SQL Server and Oracle clients) as well as ODBC drivers that enable access via ODBC to the usual Microsoft file types. The ODBC drivers to the usual Microsoft file types are redundant, since these can be opened directly. If you neglected to update your Windows system with MDAC as required in the installation instructions, please repeat the installation, install MDAC and re-install Manifold.
Troubleshooting

- Most database management system vendors selling into Windows markets will provide an ODBC driver for their database as part of the product they deliver to you. Depending on the vendor, you may or may not be able to find a freeware ODBC driver for their format if you are not their customer.

- Use a good Internet search engine to search for web sites from which you can download ODBC drivers for the less common database systems/formats.

Problems connecting to external database tables. Tables not writable or visible.

- Do you have access permission to the machine hosting the database? When accessing tables over a network your user account must have correct access permissions. Note that a local machine user account is not the same as a user account of that same name in the applicable domain.

- Is the external database correctly configured? If you cannot see tables that you know are there or if you are unable to update tables this could be a result of the configuration options specified for that database by the database administrator. Oracle databases in particular have many configuration possibilities that may prevent you from writing to or even seeing certain tables.

- Are you using non-Microsoft middleware of any kind to connect to the database? Manifold is tested using standard Microsoft connection components. If you have replaced these with third party connection components, try running with the Microsoft component and see if the problem goes away.

- Are you using ADO .NET for the connection? Tables linked from ADO .NET data sources are always read-only.

Can't Undo table operations.

- The architecture of some external database providers makes Undo operations too costly for users. In such cases you will not be able to Undo changes made to the table. In theory, Manifold could support an Undo for any change in any table regardless of the underlying data provider; however, with the architecture used some providers such an operation would require constant backing up the entire table, with unacceptable impact on performance.

Missing fields in tables. Some fields are not visible.

- Check the View - Columns command to make sure that the fields have not been hidden from view.

- Importers may have option settings that allow you to import some fields but not others. Did you import the fields you now wish to see when the table or associated drawing was first imported?

- Importing from GIS formats uses the latitude/longitude coordinate information embedded in the GIS format to draw objects. GIS newbies often expect this embedded geometric information to also automatically appear in a database table associated with each drawing. That's not how it works. See the Editing Data in Tables topic.

- Importers may skip fields containing no data. This behavior is controlled by the Import empty columns in tables option in Tools - Options. By default, the option is off, which means that importers will only import fields that contain some data.

Values in fields are missing numbers or are the wrong numbers

- Check the field types. When copying floating-point numbers to integer fields the values will be rounded to whole integer values.

- Check the formatting specified for that field. If Manifold is told to show only two digits after the decimal point, that is all that will be visible.

- Are the Area (I) intrinsic values for area unusually low in a drawing’s table? If the drawing is in Latitude / Longitude these values are quoted in degrees. Project the drawing to see them in meters.

Tables are missing records.

- Check the View - Selection Filter settings for that table. The table could be set to filter out everything except the selection.

- Check all table options. If you have told Manifold to filter out empty records they will not be visible.

- Have you sorted the table by using View - Sort or by clicking on a column head? Check the position of the record pointer by noting the position of the scroll bar. It could be that the record you were looking at has been scrolled out of view as a result of the sort.

- Are you sure you opened the desired table?
MEMO fields from a FoxPro .dbf not imported correctly.

- One problem that arises from Manifold's use of auto-adaptive .dbf reading technology is that on occasion MEMO fields written by Microsoft's FoxPro application will not load correctly. A workaround is to use ODBC or OLE DB to connect to such files instead of the built-in .dbf importer. That is, instead of specifying a .dbf file in the Files of type box when importing a .dbf file or linking to it, we would specify an ODBC or OLE DB connection in the Files of type box and then use the ODBC or OLE DB dialogs to connect to that .dbf file.

- To connect to a FoxPro .dbf file we must first install an ODBC driver for FoxPro which is part of MDAC 2.5, available at:

- To read the file via OLE DB we must install an OLE DB provider for FoxPro, available at:

Unexpected results from table commands or transform operators. Commands have no effect.

- Is the operator affected by a selection? Commands such as transform toolbar operators will have a different effect depending on whether they are run on a selection or on all objects.
- Is there a selection? If you have made a selection and don't notice that there is a selection, the operator may be working but you won't see its effect because it is being applied to only those selected records that are not in view.
- Make sure the selection is visible. If a selection is present it will affect the action of many commands and operators. Check to make sure that Selection Style is enabled so that any selected records will be displayed with a red selection color background.
- When working with tables, check the field type. It's possible that what you think is a number is really a string field, for example. The specific action of commands and tools that work with tables will often depend on the type of the field involved.
- Are you sorting on a field and getting strange sort orders? If the field is not the type you expect the sort order may be strange. For example, sorting on a text string will result in lexicographic ordering that may be different than if the same values were represented using a numeric field type.
- Are you using View - Sort and getting strange results? Check the Use Whitespace and Use Case options.

Problems with selecting records.

- Is the selection style enabled? If None is chosen then even though the records are selected they will not be shown with a red background and you might not think they were selected.
- Select records by clicking on the record handle on the leftmost margin, not by clicking on the cells in the record. This allows editing within selected records without changing the selection.
- Select records using Replace, Add and the other selection modes to add to the selection or subtract from it or other wise modify the selection.
- Make sure selected records are in view. Records may be selected but are scrolled out of view in the table.
- When using ViewBots with Rank Columns: if the "wrong" records are being selected, check the argument you are using in the ViewBot to make sure it is a fractional value such as .85 or .90 to represent 85% or 90%. Rank Columns are percentages and so should be referred to by numbers that begin with a decimal point.

Problems pasting geocoded tables as drawings

- Are you sure the table contains latitude and longitude fields for each record? A table that does not contain latitude and longitude values is not geocoded and cannot be pasted as a drawing.
Troubleshooting

- Are the latitude and longitude values expressed as decimal degrees with negative numbers (minus signs) representing West longitudes and South latitudes? If this is not the case, use the Transform toolbar for tables to convert the latitude and longitude values into the required style.

- If the points pasted from the table appear in strange positions, are you trying to paste projected tables that are geocoded with projected coordinates as latitude and longitude values? The use of projected coordinates to geocode tables is an expert activity subject to many nuances. The geocoded coordinates in the table must be in the same projection system with the same projection parameters to paste correctly into a projected drawing. Try to find the same data set in a geocoded table that uses ordinary, unprojected latitude and longitude and paste that into a lat/lon drawing in the default way.

Problems running queries

- To edit a query, right click on it and choose **Edit**. Double clicking a query will also open it for editing.

- A query cannot be run unless it first has a SQL statement placed inside it with an **Edit**.

Apparent SQL bugs

- When running SQL within Manifold queries, one is using the Manifold SQL engine. When executing SQL within the Database Console one is using whatever SQL is the native SQL of the external database system. One should be aware that SQL implementations in various database systems can contain numerous bugs. For example, even as well crafted an SQL as Jet SQL used within Microsoft's Access products contains numerous bugs. If an SQL bug occurs within the Database Console, the bug should be tracked down with the vendor of the external database system being used.

- Most "bugs" reported in SQL are in fact syntactically incorrect SQL written by the user or other user errors. If you suspect a genuine SQL bug within SQL written in the Database Console, try importing the table and launching the same SQL query. If the problem does not go away, that's a sign of possible user error since it is unlikely that the same SQL bug would appear both within Manifold's SQL and in the SQL used by whatever database system was operated via the Database Console.

- If you suspect a bug within Manifold SQL and you have access to a third party SQL (such as SQL Server or Access), try exporting the tables involved and running the same query on the external system. If the problem continues, take a second look at the SQL for possible user errors.

- Do you have Internet Explorer 6 or greater installed? Manifold requires IE installation to make sure Windows is correctly updated with scripting components, including some that are necessary for correct operation of queries.

General problems with databases

- In a perfect world, every OLE DB provider installed on our systems would work without error. Regrettably, in real life the various database drivers and providers installed in our systems will contain numerous bugs. Even in the case of drivers provided by Microsoft, one only need read the Microsoft Knowledge Base and the release notes for various Microsoft service packs to see that vast numbers of bugs have been identified in such drivers. Manifold uses Microsoft drivers and OLE DB providers to connect to various types of database tables. If there are errors in the Microsoft drivers, such errors will affect Manifold as they would any other application.

- In sophisticated applications, if you suspect a database driver bug, take time to research the issue within the online Microsoft Knowledge Base. A useful experiment might be to import the table instead of linking to it and seeing if the problem persists. If it goes away, that is evidence that the problem lies in the external database system or is a bug in the drivers for that system.

- Access permissions are a never-ending source of possible difficulties. You must be familiar with Windows access permissions as well as any permissions scheme used by your DBMS to eliminate such difficulties.
Problems with Performance
See the Performance Tips topic for advice on improving system performance.

System appears to be locked up.

- Are you sure the system is busy? Check the System Activity indicator in the status bar at the lower right hand of the screen. When the system is busy it will show a 3D "bump". It could be that the system is not busy but has been put in a command mode, such as setting a control point, where the cursor will not be freed until the chosen command is taken to completion. Press the Esc key to exit a command mode. Check command button and status line to see if any commands are enabled.
- Does the system activity indicator show the system is busy? If Manifold is busy it cannot execute a new command for you. Some commands (such as sophisticated networking analytics) can take a very long time to accomplish. See the discussion in the Performance Tips topic about scaling up tasks very slowly to avoid unpleasant surprises.
- Is another application or Windows locked up? Try pressing CTRL-ALT-DEL (in 2003, XP, 2000 and ME) to see if this unlocks the console.
- Are you using Windows '98? These systems do not have the memory management sophistication of Windows Server 2003, Windows XP, Windows 2000 or Windows ME and will routinely lock up when significant amounts of memory are used. Switch to a modern Windows edition.
- For normal operation you should have four times the maximum size of the files in use available in free disk space. If you are working with a 100 MB image you should have 300 MB free on disk. The extra space is needed to keep temporary versions of the image to allow Undo or abandoning of edits as well as to provide space for copy and paste operations.

If you can reproduce a system lockup or crash using Manifold, please send a note to tech@manifold.net with step by step instructions on how to reproduce the problem. Reproducible system hangs or crashes go to the top of our engineering priorities for immediate repair.

Unknown Errors
Manifold cannot work correctly if your machine does not have adequate RAM, an operating system that can handle the necessary RAM, or adequate disk space.

- Are you using large images? Many image formats are compressed. When imported, the images they contain will expand to require considerably more RAM than indicated by the size of the file on disk.
- Are you using Windows XP or 2000? If not and you are working with larger projects you have probably encountered a bug in Windows, not in Manifold.
- Do you have enough free space on disk? Larger projects should have gigabytes free on disk.
- Is your page file large enough? Allow 2.5 gigabytes per running instance of Manifold.
- Are you running other applications at the same time as Manifold? Other applications could be using up RAM that is needed for Manifold.
- See the Performance Tips topic for additional information.
- See the Memory Requirements topic for important information on RAM and hard disk memory requirements.

Disk Space

- Always double-check you have enough free space on hard disk. If you have no free space on hard disk, Manifold will not be able to function because there will be no room for temporary files.
- Even if you have plenty of free disk space, check the location of the TEMP folder for Windows. If the TEMP folder is placed on a disk partition that does not have enough free space, Manifold will not be able to function correctly.
- If you work with linked images, clean out your cache files from time to time. Cache files persist forever and, in extreme cases, use up your free disk space unless they are manually deleted. See the recommended actions in the Managing Cache Files topic.

Important Configuration Notice
**Very Important:** Before starting Manifold, go to the Windows Control Panel Display dialog Effects tab and make sure that the *Show window contents while dragging* check box is **not** checked.

- In Windows XP this option is found in the Control Panel's *Appearances and Themes - Display* choice under the *Appearance* tab by pressing the *Effects* button.
- In Windows XP or Windows 2003, from the Start button open the Control panel and then open the *Display* dialog. Click on the *Appearances* tab and then press the *Effects* button. Uncheck the *Show window contents while dragging* check box.
- In Windows 2000, Windows ME and Windows 98 open the Control panel and then open the *Display* dialog. Click on the *Effects* tab, and uncheck the *Show window contents while dragging* check box.
- In Windows Vista, open the Control Panel and click *Appearance and Personalization*. Click *Customize colors*, then click *Open classic appearance properties for more color options* and click the *Effects* button. Uncheck the *Show window contents while dragging* check box.

Checking this box will greatly slow down the system when displaying complex maps and images because it forces a refresh of the window contents with each minor change in mouse position while dragging the window. Make sure the box is **unchecked**.
Problems with Projections

There is no way to troubleshoot problems with projections that does not require fairly sophisticated understanding of what projections are all about. Read the Projections topics before beginning any troubleshooting. Don't skip anything since problems with projections almost always involve a battle with obscure details.

I want to combine several maps into one and the data won't line up. Objects in drawings using the same projection do not align well.

- Are you trying to combine maps using different UTM zones into one UTM zone map? That won't work due to limitations of UTM. See the Universal Transverse Mercator (UTM) topic. Get the data in unprojected form and use a different projection for a combined map.
- The same problem occurs with projected maps in Gauss Kruger, State Plane Coordinate System and many other projection systems. These are systems of projections that are designed to allow mapping of limited, local regions within a fixed system of local projections. They are unsuited for combining data from different geographic regions.
- Is the data accurate to begin with? If the drawings come from different sources it might not be a projection problem but just a matter of differing accuracy in maps from different sources.
- Are the datums the same? Using exactly the same projection parameters but different datums will result in slightly different positions that will make a difference in high accuracy applications.
- Did the data originate in unprojected maps or did you import it from projected maps? If imported from projected maps, it could be that the projection parameters you supplied are not the correct ones intended for use with this data. See Projections and Legacy Formats for discussion of the potential problems. Note that publishers of data in projected form will sometimes specify the wrong information for use with their files. The wrong datum is often a problem.
- Did you use the Edit - Assign Projection dialog in an attempt to set projections? That's a specialized dialog that should be used only to correct an error during import from a format that does not save projection info. Use the Edit - Change Projection dialog to re-project a drawing, image, surface or labels component.
- If projected data is imported from an older package, did the software used to create the imported files use the same projection standards as Manifold? Manifold uses high accuracy projection formulae and definitions to USGS standards. There is no guarantee that a legacy package uses the same definitions and high accuracy formulae. Avoid this by using only unprojected data from older packages.

Drawings, images or surfaces disappear from a map after projection.

- Did you use the Edit - Assign Projection dialog in an attempt to change projections? That's a specialized dialog that should be used only to specify the correct projection after importing from a format that does not save projection info. Use the Edit - Change Projection dialog to re-project a drawing, image, surface or labels component.
- Was the component imported from a format that does not save projection information? Formats like .dxf and .shp for drawings and .jpg or .gif in images do not save projection information. The component may look visually correct after import, but if it is not imported with the correct projection and georegistration any subsequent re-projection will cause chaos. After importing from a geographically-unaware format you'll have to manually specify the projection by using the Edit - Assign Projection dialog. See Projections and Legacy Formats.

Using a projection causes extreme distortion.

- Did you choose the right projection for your intended usage and region? Some projections cause intense distortion. Not all projections are usable for all types of data. See Projecting a Map.
- Did you specify reasonable projection parameters for your data set and intended usage? Specifying the wrong center latitude and longitude for a projection, for example, can cause even a well-chosen projection to show a highly distorted view.
- If you specified projection parameters manually after importing a projected drawing from a "dumb" format, it could be that the wrong projection parameters were provided for use with that data or that you made an error when entering the parameters. Check the projection properties to make sure they were entered correctly.
- Was the component imported from a format that does not save projection information? Formats like .dxf and .shp for drawings and .jpg or .gif in images do not save projection information. The component may look visually correct after import, but if it is not imported with the correct projection and georegistration any subsequent re-projection will cause chaos. After importing from a geographically-unaware format you'll have to manually specify the projection by using the Edit - Assign Projection dialog. See Projections and Legacy Formats.
Troubleshooting

- If you attempt to change a projection by changing the Edit - Assign Projection dialog you are not changing the projection but rather are changing Manifold's interpretation of the existing coordinate values. Change the projection using Edit - Change Projection to permanently change the projection for drawings, images, surfaces and labels. Use Edit - Assign Projection to change the projection used by a map.

**Objects drawn in maps change shape or position unexpectedly after projection.**

- Are large objects involved? When drawing an object in any geographic drawing it is important to use Segmentization to prepare the object for possible re-projection.
- Is the center of the projection (for projections that have specifiable centers) reasonably near the object? Many projections will greatly distort objects that are not near the center.
- Have you chosen a sensible projection given the location of the object? For example, using a "whole world" pseudocylindrical projection to show a single island near the Arctic circle is a poor choice if you wish to avoid distortion.
- When using the Orthographic projection, have you checked the Clip Coordinates box? If not, the entire world is "wrapped" around the projection and objects from the "opposite" side of the world will be seen co-mingled with objects from the "front" side of the world.

**Changing a projection is not permanent**

- Changing a projection in a map with Edit - Assign Projection only changes the projection used by that particular map. It won't change the native projection of components used as layers in the map. If you later open one of those components in its own window or in a different map window the projection will not have been changed. To permanently change the projection of a component used in a map, use the Project to Map choice in the layer tab context menu to re-project it to the map's projection or open the component in its own window and use Edit - Change Projection to specify any projection desired.

**Re-projecting an image or other large component causes error messages**

Manifold cannot work correctly if your machine does not have adequate RAM, an operating system that can handle the necessary RAM, or adequate disk space.

- Are you using large images? Many image formats are compressed. When imported, the images they contain will expand to require considerably more RAM than indicated by the size of the file on disk.
- Are you using Windows XP or 2000? If not and you are working with larger projects you have probably encountered a bug in Windows, not in Manifold.
- Do you have enough free space on disk? Larger projects should have gigabytes free on disk.
- Is your page file large enough? Allow 2.5 gigabytes per running instance of Manifold.
- Are you running other applications at the same time as Manifold? Other applications could be using up RAM that is needed for Manifold.
- See the Performance Tips topic for additional information.
Problems with the Internet Map Server

See the Map Server Overview for general information on Manifold IMS (Internet Map Server). This topic covers some of the same points in slightly different ways since often one of these points has been overlooked.

Read through this entire topic at least once, then print it out in hard copy and go through it line by line, checking each line off with a pen or pencil as you investigate. Take nothing for granted and assume nothing. For example, don't assume permissions are correctly set. Actually check each and every file and folder involved for the right permissions on the IUSR_ or NETWORK SERVICE or ASPNET account, as the case may be.

Getting Ready

Running Manifold IMS has to do mostly with elementary operation of Windows as well as Microsoft's Internet Information Server (IIS) or whatever HTTP server is being used. Running Manifold IMS itself is very simple.

Of course, IMS can't run on a web server machine if Manifold has not been installed and activated on that machine.

In all cases of debugging, login using the Administrator account on the machine being used as a web server. Use the actual, local Administrator account (which may need to be enabled for usage in Vista) and not just some account that you are sure has Administrator privileges, whether it be a local account or a domain account. Use the actual, local Administrator account.

Tech tip: Users (especially in Vista) who are convinced the account they are using has "adequate administrative privileges" should be the first to login using the actual, local Administrator account.

To run a web site you need basic Windows administration skills. You must understand what is mean by a login and what is meant by permissions and how to check and set permissions. Any good book on Windows administration will explain this. If you do not know what permissions are and how to check and set them, you will not be able to identify and eliminate the most common user errors that prevent IMS web sites from running.

Verify Manifold installation

The first step in case of trouble is to verify that Manifold has been installed and activated on the web server machine. This is easy to do. Gain physical access to the web server machine and login using the Administrator account. Stop IIS. Reboot the machine. Verify IIS is still stopped. Now, launch Manifold interactively. Does Manifold come up without any activation dialog? If yes, you have Manifold installed and activated on the web server machine. Check the Help - About dialog and verify that you are not running Personal Edition (Personal Edition does not have IMS). While you have the Help - About dialog open, write down the exact build number in use, the Manifold edition in use and any extensions that are installed. Exit Manifold. Restart IIS.

If you have installed a Manifold edition other than a Runtime license and you cannot launch Manifold interactively, then you have not correctly installed Manifold and Manifold IMS will not work.

Note that the above verification is not possible if you have installed Manifold using a Manifold Runtime license. Runtime licenses cannot be launched interactively. The lack of an ability to launch Manifold interactively for debugging purposes is one reason that runtime licenses are not recommended for IMS beginners.

Verify installation of accessories

Some web sites may involve accessory software, either Manifold extensions or software provided by third parties. Check to make sure any such software has been installed. The most common errors:

- Verify presence of required Manifold extensions. If your web site utilizes features from Business Tools, Geocoding Tools or Surface Tools make sure these have been installed and activated. You can verify this in the Help - About dialog that was opened in the above paragraphs.
- Verify presence of image server modules, if used. Many web sites use Image Servers to fetch background images. Make sure that the web server machine has the required modules installed (usually placed in the Manifold installation folder). It is a very common error to create a project on some development machine using image server modules and then to forget to install the required image server .dll files on the web server machine as well.
Troubleshooting

- Verify presence of other required software. If your web application uses other software, such as DBMS packages or other software, make sure the required applications have been installed. For example, if you have created a web page using Excel web queries as shown in the A Flashy Demo - Web Queries and KML example topic, you will need to have Microsoft Excel installed on the web server.

Verify web server operation

The next step in case of trouble is to verify that your machine is capable of operating as a web server and that you know how to create and operate simple web pages on that machine. This is not something to be taken for granted, especially if other people have had access to your web server machine, as it is possible that some person or security update has altered IIS or other Windows settings in a way that prevents basic web server functions from operating as you expect.

Before debugging any problems with Manifold IMS, please create a simple web site not using Manifold IMS on your web server and browse it from a different machine. If you cannot create a web site on your web server that does not use Manifold IMS, then most likely the problem (or, at least one of the problems) has nothing to do with Manifold IMS. Get your web server running first with a simple web site.

For example, if you are running Windows Server 2003 you might discover that no web site using .asp pages can run.

Important: For enhanced security, Microsoft's Windows Server 2003 product comes with Internet Information Server configured so that .asp pages are not enabled. .asp pages must be turned on within IIS for Manifold IMS to function correctly.

When running Manifold IMS in 64-bit Windows editions, use of 32-bit ASP / ASP .NET must be enabled. See the discussion in the Creating a Web Site topic.

Create a Simple HTML Page

In your web page publication folder, create a file called hello.html that contains the following text:

```html
<html>
<body>
<p>Hello, world!</p>
</body>
</html>
```

Can you view this file as a web page from your local machine? Can you view this web page from your intranet or Internet as you would like to see your IMS site? If you cannot view this file, do not proceed any further with IMS until you can figure out how to create, publish and view this simple web page.

Create a Simple ASP Page

Once you know you can create and view a simple HTML file you know that your web server is operational. The next step is to verify that it can handle ASP by creating and viewing a simple .asp file.

If you are creating IMS pages using VBScript, create a file called hellovb.asp that contains the following text:

```html
<html>
<body>
<% response.write("Hello, world!") %>
</body>
</html>
```
If you are creating IMS pages using JavaScript, create a file called `hellojs.asp` that contains the following text:

```html
<%@ language="javascript"%>
<html>
<body>
  <% Response.Write("Hello, world!") %>
</body>
</html>
```

Can you view the `.asp` file as a web page from your local machine? Can you view this web page from your intranet or Internet as you would like to see your IMS site? If you cannot view this file, do not proceed any further with IMS until you can figure out how to create, publish and view this simple `.asp` web page.

Create a Simple ASP .NET Page

ASP .NET is not enabled by default in recent versions of IIS. To verify ASP .NET is up and running create a file called `test.aspx` in your root web server directory and paste the following into it:

```html
<html>
<body>
  <h1>The date is <%= Response.Write(DateTime.Now.ToLongDateString()) %></h1>
</body>
</html>
```

Visit `test.aspx` with a browser and if you see a sentence giving the current system date ASP .NET is working.

[This example was contributed by user mlinth on the Georeference forum, and comes from the book *Beginning ASP.NET 3.5 in VB 2008: From Novice to Professional, Second Edition* by Matthew MacDonald.]

Next Steps

The Manifold side of troubleshooting consists mainly of checking that Manifold is correctly installed, is available to all users, and that the files needed to be used are where the `config.txt` file says they are. Check the following:

- Has Manifold Professional Edition or higher been installed on the web server machine? Note that Manifold Personal Edition does not include IMS.
- When Manifold was installed on the web server machine, did you login as Administrator to do the installation? Note that Vista may require the Administrator account to be enabled before you can use it.
- When Manifold was installed, was it installed for use by all users?
- If it is more than 30 days since the serial number was issued, has Manifold been permanently installed using an Activation key on the web server machine?
- When Manifold was activated, did you login as Administrator when entering the Activation key?
- Are you attempting to run Manifold simultaneously in interactive mode and as the IMS map server at the same time? Execute an `iisreset` when changing from interactive use of Manifold to IMS use or vice versa.
- Have you copied all the files created by the Export Web Page dialog to the correct folder within the `Inetpub\wwwroot` folder hierarchy on the web server machine?
- Has the `.map` file being used been placed at the location specified for it in the `config.txt` file?
Troubleshooting

- If you are using ASP, does the system's IUSR_ user login have permission to execute Manifold.exe in the Manifold System installation folder? Check this using Windows Explorer to view the security settings on the file.
- If you are using ASP, does the system's IUSR_ user login have permission to read the .map file in use? Check this using Windows Explorer to view the security settings on the file.
- If you are using ASP, does the system's IUSR_ user login have permission to read and write the system's TEMP folder? Check this using Windows Explorer to view the security settings on the file.
- If you are using ASP, does the system's IUSR_ user login have permission to read and write all folders and files used in the application, such as temp folders for linked images, .mdb or other files used in linked tables, or database access for components linked from databases?
- When working with Windows Server 2003, have .asp files been enabled in IIS?
- When working with Windows Server 2003, does the NETWORK SERVICE account have all the permissions enumerated for the IUSR_ account above?
- When working with Windows XP x64, does the NETWORK SERVICE account have all the permissions enumerated for the IUSR_ account above?
- If you are using ASP .NET, does the ASPNET account have all the permissions enumerated for the IUSR_ account above?
- If you are working in a 64-bit Windows edition, have you installed a 32-bit version of the .NET framework as well as the 64-bit .NET framework?

The number one problem with IMS reported to tech support is that users neglect to add the IUSR_ account with access permissions to security settings for the .map file in use. A related error is that after checking permissions are correct users edit the .map file with Manifold and, without realizing it, change permissions so that the IUSR_ account no longer has access permissions to the .map file. Saving a .map file with Manifold will preserve file permissions on Windows 2000 and later systems, but this is always something worth checking.

To check permissions, using Windows Explorer (do not just depend on the IIS management console or other server management console), right click on the .map file, choose Properties and verify in the Security tab that the IUSR_ account for the system has necessary read and execute permissions.

Very important: In 32-bit and 64-bit versions Windows Server 2003 and in 64-bit Windows XP the account used for Internet visitors is not the IUSR_ account as with earlier Windows versions. Instead, the NETWORK SERVICE account is used. When ASP .NET is employed the ASPNET account is used.

Windows Server 2003 and 64-bit Windows XP operators should understand the NETWORK SERVICE account is meant in all cases where this documentation discusses the IUSR_ account.

Web pages operating in an ASP.NET environment may require different permission settings than those operating in an ASP environment. For example, the default login used by ASP.NET is ASP.NET, while the default login used by ASP is either the IUSR_ account or the NETWORK SERVICE account. When running in ASP .NET, the .map file and all other data being accessed from the ASP.NET environment must be readable by the ASPNET user account.

ASP .NET operators should understand the ASPNET account is meant in all cases where this documentation discusses the IUSR_ account.

While there is no end of trouble that can arise from willful misconfiguration of Microsoft Internet Information Server when attempting clever tricks, in the case of simple web sites problems arise most frequently from a handful of common errors:

- Can you create and browse a simple web page not involving Manifold IMS on the web server machine? If you have not correctly configured or installed Microsoft IIS on the web server machine so that you can create simple web pages without using IMS, then you will not be able to serve web pages involving Manifold IMS either.
- Attempting to run a web site with Manifold IMS at the same time Manifold is running interactively on the same system. Do not run Manifold interactively at the same time you are trying to run IMS. To make sure Manifold is not being run interactively, reboot the web server machine after running Manifold interactively.
- Failure to arrange required read and execute privileges for the IUSR_ account (or, the NETWORK SERVICE account or ASP.NET account if applicable). The IUSR_ account must be able to access the .map project as well as any tables required for linked tables or linked drawings and it must be able to execute Manifold System. The IUSR_ account must also have read and write privileges in the system TEMP folder as well as execute privileges for the Manifold.exe executable. In case of trouble, always check the permissions on the .map file. Use Windows Explorer to check all permissions: don't rely
on the IIS management console to check permissions. If any linked components appear in the project, the IUSR_ account must have read privileges to work with those files as well. For example, if an .mdb file is used to generate a linked drawing the IUSR_ account must have read and write permissions for that folder so that the lock file for the .mdb file can be created. When checking permissions, open Internet Explorer, right click on the file or folder, choose Properties and then check the settings in the Security tab. Don't assume that the web administration dialogs used by IIS will set these permissions for you. If using image servers, the IUSR_ account must have read and write privileges in the folder used to cache the image server files.

- If you have opened the .map file in Manifold System, check the permissions again. They might have been resent to not include the IUSR_ account. Saving a .map file with Manifold will preserve file permissions on Windows 2000 and later systems, but this is always something worth checking.
- Failing to install Manifold System on the web server machine for use by all users (a checkbox in one of the Manifold System installation dialogs).
- Neglecting to maintain links correctly when working with a project that uses linked tables, linked drawings or Enterprise Edition components.
- Attempting to use Enterprise Edition features, such as shared components, on a web server machine that is not licensed for Enterprise Edition.
- Failure to install the Geocoding Tools extension as well as the geocoding database within the Manifold System installation folder when using queries that include geocoding functions (or failure to have MapPoint installed if it is being used as an alternative geocoder).
- Failure to install or activate (using an Activation key) features contained in the Business Tools, Geocoding Tools or Surface Tools extensions if these features are used by the IMS application.
- Neglecting to place the .map file where the config.txt file says it is supposed to be.

**Manifold or the map server won’t launch**

- To run the map server and serve web pages, no interactive session with Manifold can be running on the same machine.
- To work with Manifold interactively, no map server session can be running on the same machine.
- Do an iisreset command in the Windows Command Prompt window before attempting to work with Manifold interactively after the map server has been active.
- Make sure to save the .map file after creating it or making any alterations interactively in Manifold. The .map file must be saved before the map server can use it. After editing a .map file, make sure by using Windows Explorer to view Security settings that the IUSR_ account (or, the NETWORK SERVICE account if applicable) still has read permissions on the .map file.
- Have you provided a serial number and Activation key? When Manifold is running as a map server it will not raise the Activation dialog. Before attempting to use the map server within Manifold you must launch Manifold interactively at least once to provide a serial number for preliminary installation, or a serial number and Activation key for permanent installation. See the Activation Keys and Serial Numbers topic. If you provided just a serial number without an activation key and it has been more than 30 days since the serial number was issued, you will have to launch Manifold System interactively so you can provide a serial number and an Activation key for permanent installation.
- When installing Manifold, did you install Manifold System for everyone to use on the computer? This is the default choice in the installation dialogs.
- When you installed Manifold System, did you login as Administrator to perform the install? If you activated Manifold System using a serial number and Activation key, did you login as Administrator to do so? It is a common error to think that using an account which supposedly has "Administrator" rights is an OK substitute for using the actual Administrator account. Avoid unpleasant surprises by using the actual, local Administrator account.
- Check access permissions on the .map file as noted below.
- When using a .map project using Enterprise Edition features, the web server must have Manifold System Enterprise Edition installed.

**Permissions**

- Note that in Windows Server 2003 or 64-bit XP the account used for Internet visitors is not the IUSR_ account as with earlier Windows versions. Instead, the NETWORK SERVICE account is used. Likewise, when running ASP .NET the account used is ASPNET.
- The .map file used with the web site must have security permissions such that the IUSR_ account (or, the NETWORK SERVICE account or ASPNET account if applicable) used for Internet browsers has read access. Manifold must be installed for everyone to use on the machine. If any linked drawings or linked tables are used in the project that is published, the IUSR_ account must have all necessary privileges required to work with the tables that control those linked drawings or linked tables. Note: the
Troubleshooting

"IUSR_*" account is named using the computer name of the web server machine. If the web server machine is named "GODZILLA" this account will be called "IUSR_GODZILLA".

- When using linked images in an IMS web application it is important that the web application has read and write permissions in the cache folder, or otherwise it will not be able to use the tiles cached for the linked image.

General problems browsing the web site

- Confirm IIS is operational by launching Internet Explorer and browsing http://localhost ...If this does not work, IIS is not functioning. You must have IIS functioning correctly before attempting to work with Manifold IMS within IIS.
- Verify the directory to which the Manifold IMS project is published is within the publication directory structure of IIS. Create a small test.html file and then browse to it using http://localhost.pathname/test.html where "pathname" is the directory path to the directory in which you are placing your Manifold IMS files.
- Check to make sure Manifold is not running on the system interactively. Manifold IMS cannot function on a machine where an interactive Manifold session is running. Close all Manifold sessions, reboot and try the browser again.
- Make sure no interactive Manifold sessions running on different machines are trying to use, across a local network, the .map file from which the Manifold IMS site was published. Manifold IMS needs to use that .map file during operation and no other process should use that .map file.
- If a newer version or service pack of Manifold has been installed, or if you have worked with Manifold interactively on the web server machine, reboot the system to make sure there are no older processes lingering.
- After changing either the .map file in use or any of the .asp files or accessory files in the Manifold IMS site, do an issreset in the Windows Command Prompt window to unload all objects involved so that the new versions are guaranteed to be in use.

Browser compatibility

- Client browsers must be at least "4.x" vintage IE, Netscape, Mozilla or Opera browsers when the "Compatible" template is used to generate a site.
- The Info button is not available in sites generated using the "Compatible" template.
- Client browsers must be at least "5.x" vintage IE, Netscape, Mozilla or Opera browsers when the "Standard" templates are used to generate a site.

Configuration problems

- When developing a Manifold web site on one machine and then installing the site and .map file on a different machine that is used as a server, make sure that the server machine is correctly updated with the correct .map file.
- Make sure the server machine has the latest version of Microsoft Internet Explorer installed (at least IE 6 is required).
- Make sure the server machine is running Windows Internet Information Server 5.0 or more recent.
- Check that the server machine has the same version of Manifold installed together with the most recent service pack (if any have been issued) as the development machine. For example, if your development machine uses Manifold 8 and the web server is still running Manifold 7x, the web server machine won't be able to open the Release 8 version .map file.

Queries don’t work as expected

- Stop IIS in Internet Services Manager. Execute an issreset in a Windows command prompt window. Launch Manifold interactively on the server machine and see if the query works correctly.
- Check the component that is published (it will be listed in the config.txt file in the IMS web site files). Does the query in any way refer to a component outside that which is published?
- Is the query a SELECT query? Only SELECT queries are allowed.
- Does the query generate more than 200 records for the table? Only the first 200 records will be published to the web page.
- Does the query use features available only through extensions? If so, has the requisite extension been installed?
A site that worked before no longer works

- Check to make sure that IIS is running.
- If IIS is running and the browser shows an error message citing a particular .asp file and a line number, check to make sure there is not an interactive session of Manifold running on the server.
- Make sure Manifold has not been removed (de-installed) from the web server.
- Using Windows Explorer check permissions for IUSR_ on all files and folders involved. Note that re-editing a .map file can change permissions on that file.
- Check to make sure that the Manifold installation has not been damaged (e.g., the Manifold directory in the installation has accidentally had files deleted, etc.)
- Perform disk maintenance to make sure disk errors or other problems have not damaged the Manifold installation.
- Verify there is adequate disk space for all temporary files created by Manifold IMS.
- Reboot the system to fix any Windows instability.
- Did you activate your copy of Manifold? It could be more than 30 days since your serial number was issued and an Activation key is now required as well as a serial number.
- Make sure no interactive sessions of Manifold are running on the web server.

If geocoding functions do not work within IMS check the following:

- Manifold Geocoding Tools extension has been installed.
- The Geocoding Database has been installed into the GCDB folder within the main Manifold installation folder. By default, this location is C:\Program Files\Manifold System\GCDB.
- All state files required for the geocoding database are present (that is, required files have not been removed after installation).
- If MapPoint is to be used as an alternative geocoding data source, has MapPoint been correctly installed? Note from the main IMS topics that MapPoint requires special permissions settings.

More performance is needed

- Read and think carefully about every line of the Performance Tips topic.
- Read and think carefully about every line of the Optimizing Performance topic.
- Eliminate any unused fields in the components being published.
- Eliminate any unused components in the project being published.
- Make a backup copy first, and then simplify the drawings used by increasing the size of the View - Properties - Precision parameter and then running Normalize Topology. Most drawings used in web applications can be reduced in size/complexity by a factor of ten without any objectionable visual effects. This will dramatically speed up the site.
- Don’t use images when drawings will do.
- If you must use images, use linked images that are compressed images or linked images served from an Oracle Spatial server or other DBMS server.
- Make sure that all components appearing in a map have the same native projection, which should be the same as the map’s projection.
- Use a smaller size window in the IMS site. Users may be complaining because you are sending very large images to them that take time to download through a slow Internet link.
- Use x64 Windows and x64 Manifold.
- Add RAM memory to the system. RAM is cheap. Install gigabytes of it. Install enough so the entire .map file plus all of your Windows processes can be in RAM.
- Use large, fast hard disks spinning at 7200 RPM or faster.
- Use SATA interfaces on your disks and configure a striped RAID array.
- Specify a static (predefined size) Windows page file that is far larger than ever will be necessary. This is faster than a dynamically re-sized page file.
- Use multiple hard disks. If you have two disk drives keep the system TEMP directory on a different disk drive than the web site. Ideally, if you have four disk drives the system TEMP directory, the system page file, the .map file and IIS and the web site will all be on different disk drives. This allows the disk drive heads to seek independently from each other as each of these four different files is used. This is
Troubleshooting

less of a factor if you have so much RAM that the page file or the .map files do not require any disk accesses.

- Use faster processors or use multiple, multicore processors.

Problems with ASP .NET

- Did you choose an ASP .NET version of the templates in the Export Web Page dialog?
- ASP .NET allows use of dependent DLLs, which if the system is misconfigured will lead to error messages such as "A dynamic link library(DLL) initialization routine failed." The most likely reason one or more DLLs are not getting initialized when loaded in an ASP.NET environment is that the system can not locate the dependent DLL modules to which it is linked. To work around this, include the path to the Manifold installation folder into the system PATH variable. That is, go to Control Panel - System - Advanced - Environment Variables and append the path for the Manifold installation folder (typically, C:\Program Files\Manifold System) to the system (not user!) PATH variable. Then restart IIS, or, better yet, reboot the machine. This should get rid of the problem.

- Web pages operating in an ASP.NET environment may require different permission settings than those operating in an ASP environment. For example, the default login used by ASP.NET is ASPNET, while the default login used by ASP is IUSR_xxx, where xxx is the machine name. The .map file and all other data being accessed from the ASP.NET environment must be readable by the ASPNET user account. Use Windows Explorer to check the security settings for all files involved in your web site to make sure that ASPNET can read them or execute them (in the case of executables such as manifold.exe). See the analogous discussion for the IUSR_ account.

See Also

Map Server Overview
Creating a Web Site
IMS Config.txt Options
IMS Queries
Optimizing Performance
Performance Tips
Publishing Multiple Pages
Technical Support

Most Manifold System users find that the Help documentation, web pages, training aids and participation in Internet forums amply serve their needs for understanding and support. For those users who want direct technical support by manifold.net staff, technical support products are offered for sale by manifold.net, including options for email technical support as well as voice telephone support.

License fees paid for Manifold System do not normally include technical support. Although manifold.net may provide technical support (including Service Packs or some limited number of support incidents, typically two per license) at no charge on a voluntary basis, the voluntary provision of technical support is not a commitment by manifold.net to continue to provide technical support free of charge. Normally, any free incidents that are provided as part of a license are standard tech support incidents that cover installation and interactive features only. Standard support incidents do not cover developer level features such as scripting, programming, customization, IMS, Enterprise Edition features, SQL and any question involving Runtime licenses.

To learn about technical support products or services (either paid or free) that might currently be available for Manifold System, please visit the Support page at www.manifold.net - this page will always be updated with the latest technical support information for Manifold System. Information and procedures published in the Support page on the website supersede this topic.

Regardless of how you acquired your Manifold System license or what technical support arrangements you have made, the Manifold team always appreciates receiving direct email with bug reports, comments, or suggestions for product enhancements. For tips on making suggestions that will rapidly influence the product design process, please see the Contacting manifold.net topic. To avoid spending a support incident, make sure that any such suggestions or reports clearly indicate that you do not want a reply.

Very Important: Please do not send any proprietary information or suggestions in which you or any one else claim any intellectual property interest. Please only send information or suggestions that are in the public domain.

Free Support Resources

The fastest way to get technical support is to not need it in the first place. The following resources are free to all users:

- Search this documentation thoroughly before trying alternative routes to support. Most requests for technical support are answered with a one-line email citing the relevant topic in this documentation.
- Read the Troubleshooting topics.
- Check the Support page at www.manifold.net to see if any routine updates for Manifold System have been released. If so, download and install the latest update. Updates add new features and repair bugs. New features will often remove limitations or otherwise make the system easier to use.
- Visit pages at the manifold.net website that give examples of customization (.xml files, for example), scripting within Manifold, programming in third party applications, web site creation and so on.
- At the www.manifold.net website, visit the Support page to consult the Support FAQ and other free resources.
- Participate in the Manifold System user community online forum at the GeoReference forum, especially if you are engaged in scripting or other advanced uses of Manifold. See the Help - Manifold on the Web topic for fast access to the Manifold online community and other resources. See the Support page on the Manifold web site for links to other online user communities.
- As always, Internet links can change overnight. If you can't find the above forums or links, see the Support page for current information.

Note that posting a question to an online user community such as the GeoReference forum is not the same as communicating directly with manifold.net technical support or sales personnel. If you wish to report a bug, make a suggestion or request technical support, contact manifold.net directly.

Do Not Waste Support Incidents

With the exception of initial installation and activation questions, any question sent to technical support or any question of a technical nature, however trivial, sent to other manifold.net addresses will cost an incident.
While it is entirely up to the user what questions he or she wishes to ask of technical support, the advice of the tech support staff is to avoid wasting technical support incidents on utterly trivial questions that are easily found in the documentation or other free resources. Most questions sent to technical support are elementary questions already answered in Help, the Support page or other manifold.net web pages or on free Internet resources such as the GeoReference forum. The wise user will avoid wasting support incidents on easily-found information.

See the Help - Manifold on the Web topic for fast access to the Manifold online community and other resources.

**All Support Questions are Routed to Tech Support**

Tech support incidents also apply to all questions of a technical support nature that are sent to sales or to any other manifold.net email address. (Some users occasionally try to gain additional incidents by sending email of a technical support nature to sales. Such emails are forwarded to tech support.) This policy applies to all questions, no matter how trivial or sophisticated. Do not waste support incidents by sending simple technical questions to sales or other manifold.net email addresses. Instead, take a moment to search and read Help.

Some manifold.net staffers participate in various Internet forums. All such participation is voluntary and is done on the employee's own time, not on company time. If you obtain the email address of a manifold.net employee, please do not use that address to send questions of a technical support nature to that employee. All such questions will be routed to tech support and a tech support incident will be required. This policy is intended to protect the individual privacy of manifold.net employees and to preserve the ability of staffers to participate in public forums without risking abuse from unfairly directed tech support questions.

**Using Support Incidents / Tokens**

If you have received some limited number of support incidents as part of your purchase, visit the Support page at the web site to learn how to request technical support.

Tech support can only be obtained by using technical support tokens. A token is like a serial number, a sequence of letters and numbers that uniquely identifies the rights to one technical support incident. Tokens are available for standard support incidents and for developer support incidents. Tech support inquiries (regardless of which manifold.net email address they are sent to) unaccompanied by an unused technical support token will not receive technical support.

There are two ways to get technical support tokens:

- You may receive some number, usually two, free tokens as part of acquiring a Manifold license.
- You may purchase technical support tokens. Purchasing a technical support project provides you with tech support tokens of the number and class you have purchased.

**Keep your tokens confidential.** Whoever has possession of a valid token can use it for tech support. If someone else obtains a support token from you they can use it to get support and thereafter that token will be used up - you will not be able to use it.

To use a support token, write to tech@manifold.net with the token included in the first few lines of your email. Include that token in any further email exchanges related to that incident. For example, if tech support asks any questions that require a reply from you, include the token in your reply. A convenient way to do so is to simply include the body of previous email exchanges in the text body of your reply.

**Do not use attachments or HTML email.** Any attachments will be stripped out for security reasons, and HTML email will be converted to text email.

**General Guidelines for Contacting manifold.net**

**Very important:** When emailing manifold.net, to avoid having your email deleted by anti-spam software, do not attach any files, cc any third party, or send emails that have blank subject lines, blank "from" fields or almost blank subject lines such as "Re: " with no other subject text. We suggest using a subject line that clearly identifies your email as a matter that involves manifold.net or Manifold System.

Do not use obscenities or words commonly found in spam, such as references to drugs or other off-topic matters. This will avoid having your email deleted by automatic spam filters.
Other Guidelines

When interacting with manifold.net, please observe the following:

- Please include all of your comments within the email. Do not attach .doc, .pdf or other files containing text. Do not send images unless requested.
- Please do not attach any files to emails sent to tech support. If a sample file is requested, tech Support will provide upload instructions to upload the files by FTP.
- Do not contact technical support unless your system is up to date with the latest Manifold edition, the latest Manifold update, all current Windows service packs and other Microsoft updates.
- Please respond promptly to tech support emails. Incidents inactive or involving tech support questions that are left unanswered for seven days will be closed. Re-opening the matter later will cost another incident.
- Please answer all questions asked by tech support. If you do not answer all questions the incident will terminate. Re-opening the matter later will cost another incident.
- Keep in mind that there are many engineers at manifold.net, so the person who replies to some subsequent exchange in a thread might not be the same person with whom you initially corresponded. That's why it is a good idea to preserve the entire thread within any reply you make.

Please follow the above guidelines to avoid wasting tech support incidents. Tech support can help you rapidly if you provide full information and promptly answer all questions.

Runtime Licenses

Runtime licenses are provided with no bundled support incidents. Any questions involving a Runtime license require a developer support incident. If you plan on using a Runtime license and deploying it to a different user, plan on providing any support that user requires.

Keep Up To Date

Do not contact technical support unless you have:

- Installed the latest release of Manifold and the latest update for that Manifold release.
- Assured your computer system fulfills all hardware and software requirements set forth in the Requirements web page for your edition of Manifold.
- Installed the latest Windows service pack for your version of Windows.
- Installed the latest Jet 4 service pack.
- Installed IE 6.0 or greater.
- Installed the latest drivers for your video card and printer if these are involved.

Manifold uses many Microsoft system resources. If there is a bug in Microsoft code, it could cause problems within Manifold. Microsoft has a continuous program of identifying and fixing any bugs in Windows or other Microsoft components through service packs. In addition, Manifold service packs will fix bugs that are discovered within Manifold.

Therefore, a condition of using tech support (to avoid wasting tech support resources on problems that have already been fixed) is that you must have the latest Manifold release and update installed. You must also have the latest Microsoft service packs for your version of Windows, for IE and for Jet 4 installed. You must have IE 6.0 or greater installed as well.

Tech Support Offered Only for Current Release

Tech support is available only for the most recent release and the most recent update of Manifold System. If a new release of Manifold System is announced make sure to upgrade to the new release if you would like to use technical support incidents. An important part of every update and new release is the correction of bugs found in previous releases. Having the most recent release and update of Manifold installed avoids any time wasted on the consideration of bugs that have long been fixed. It also avoids confusion raised by old methods that have been superseded by newer and better ways of operating Manifold.
Whenever a new release of Manifold System is announced, licensees of the previous release have traditionally been offered an upgrade for a nominal fee for a limited time (usually 45 days after publication of a new release). After that, the fee for upgrading could be considerably higher so don't miss the opportunity to upgrade to the new release when it comes out.

It is especially important to upgrade to a new release if you have procured technical support products such as a ten-incident support product. Any unused tokens from the previous release of Manifold will be wasted if you do not upgrade to the most recent release. It makes most sense to take advantage of the initial, nominal cost, upgrade offer when it comes out.

If you actively use Manifold, by default Manifold itself will notify you when a new update (including new release) has been published. If you are not actively using Manifold, make sure to check into the manifold.net web site from time to time for news of new updates and new releases.

Do Not CC Third Parties

When writing to tech support, do not "cc" third parties. Manifold.net tech support does not respond to emails that are shared with third parties or are cc'd to any other recipient, including postings on newsgroups. There are three reasons for this policy.

- The first reason is financial: A tech support incident is a valuable service provided to a single individual. It is not offered to multiple individuals at the same time.
- The second reason is practical: Experience shows that users will puff up their expertise and are less willing to admit dumb actions when an exchange takes place in front of third parties. To avoid any incentive not to be completely honest all tech support exchanges must occur in private.
- The third reason is privacy: Tech support may at times need to discuss information that is covered by the Manifold privacy policy. Because the policy allows no disclosure to any third party we require all such communications to occur in private. Even if users waive their privacy rights we are not willing to waive our privacy policy. Only if it is never violated can we absolutely guarantee the privacy policy.

Support questions that are cc'd to third parties may end up wasting any tokens used.

Contacting Technical Support using a Token

Tokens are long alphanumeric strings that are sent to you when you purchase technical support incidents, like the following:

63AD4E38E989-4490B540-7DF13F3D3ACDCF56EA6CC8896CC9

If your email viewing program uses a narrow window, when you read the email that sends you your tech support tokens you may see the a token as "wrapped" to appear on more than one line. There are no paragraph breaks or "end of line" characters in a tech support token: it all consists of letters and numbers on a single line.

When using a technical support token, provide it exactly as sent to you. Do not change upper case to lower case, add or delete any numbers or letters or hyphens. Do not change it in any way. To avoid confusion between digits such as zero and the letter "O", it is wise to use Copy and Paste to copy the tech support token from the email in which it was sent and then to paste it into the email you are writing to tech support.

If you have a valid, unused tech support token you may contact manifold.net technical support by sending an email to tech@manifold.net - You must include a valid, unused technical support token in the first few lines of your email message. The example below shows the beginning of an email letter to tech support that begins with a token on the first line and then continues with the letter:

63AD4E38E989-4490B540-7DF13F3D3ACDCF56EA6CC8896CC9

Dear Tech Support,
I am using 8 pro (8.1.0.1834) on WinXP Pro SP3 running on a Dell Latitude 1300 laptop with a Core 2 Duo, 2GB of RAM, 60GB of disk with 25GB free
space. When I open a map window that contains a drawing layer (it can be any
drawing, like the example Mexico drawing on the download site) and then
choose File - Export - Drawing the only choices I see in the Save as type box
in the Export Drawing dialog are DXF files or KML / KMZ files. How do I
export to shapefiles?

Technical support and the email "bots that automatically scan all emails to tech support know a token when they
see one, so there is no need to add descriptive text such as "I'd like to use the following token" or other
comments. Simply placing your valid, unused technical support token in the first few lines of the email message
is enough.

Include the token in any subsequent emails in that thread. This is easily done if you get in the habit of including
all previous exchanges in the body of any response.

If you ask a question that requires a developer support token (any questions on programming, scripting, SQL,
Enterprise Edition, web serving, any customization of any kind) you must provide a developer level support token.
A free token for standard technical support will not suffice.

If you do not have a technical support token, you cannot use manifold.net tech support. If you send an email to
technical support without including a valid, unused technical support token in the first few lines of the email
message, you receive a form letter reply indicating the need for tech support tokens. If you use a token that has
already been used up, you will receive a form letter stating the token you used has already been used up.

If you send an email asking questions of a technical nature to any other manifold.net email address it will be
routed to technical support. Folks who repeatedly write with technical questions to other manifold.net email
addresses (such as sales) will have whatever tokens they have available used up, and could end up being barred
from manifold.net email systems.

Information Required when Contacting Tech Support

Any request for technical support must include, at a minimum, the information technical support may need to
service the request. Such information must include:

- A valid, unused technical support token.
- The specific manifold.net product you are using with build number from the Help - About dialog.
- The exact Windows version in use and the Windows service pack applied.
- The exact version and name of any Windows emulator, virtual machine or any other operating system
  modification, if in use.
- The brand of personal computer you are using (or motherboard).
- The processor you are using.
- The amount of RAM installed in your system.
- The total amount of disk space installed in your system and the total amount of free disk space.
- A description of the problem.
- A description of the data set or map you were using.
- Very Important: If you can do so, provide a specific sequence of actions that reproduce the problem
  using only maps or data sets from the Manifold CD.
Submitting a request for tech support without the above information will delay assistance until the required information can be obtained. See the example above.

**Tech tip:** The answer in the above example is to open the drawing in its own window to be able to export to the full list of formats to which drawings may be exported. Alternately, right click on the drawing in the project pane and choose Export. Maps have layers, so when opening a map window and choosing File - Export the map can only be exported to those drawing formats available for export that work with multiple layers, such as DXF or KML / KMZ.

### Checking Status of Tech Support Tokens and Serial Numbers

To check the status of any particular technical support token or Manifold serial number, visit the Support page on the www.manifold.net web site to find a link to the Serial Number Status page. The status page may be used to find out if a particular technical support token has already been used.

The status page also may be used to find out if a particular serial number is a 32-bit or 64-bit serial number, whether it is still active or has been revoked (as might occur if it has been traded in for an upgrade), what product it authorizes and how many Activation keys are available.

**A reminder:** Any use of a serial number or technical support token, such as checking status via the Serial Number Status page, should be done using Copy and Paste from the original serial number email sent out by manifold.net.

When checking the status of Manifold serial numbers, do not use the "masked" version of the serial number that may be displayed in the Help - About dialog which ends in a series of X characters. The masked version displayed in Help - About has had the final characters altered with a series of X characters so that someone who has physical access to your computer cannot steal your Manifold serial number. The masked version displays enough of the serial number so you can determine for your internal record keeping which serial number you used on a particular machine, but not enough of the serial number for someone to be able to steal it and use it to obtain Activation keys or other wise use it.

### Bug Reports

User reports are a key aspect of identifying, tracking down, and annihilating bugs. The most important requirement for this process is the ability to reproduce a reported bug in the lab. Therefore, we ask that any bug reports (if at all possible) include a specific sequence of steps that will reproduce the bug when applied to a data set that is provided on the Manifold download site. Please make sure you have installed the latest Service Pack and have verified the bug still exists before submitting a bug report.

Bug reports are **not** considered incidents if you do not request a reply. If you wish to report a bug, send an email of any length; however, make it clear you do not require a reply. For example, begin your letter with "Bug report - no reply necessary."

Users will at times send in a bug report and then ask that the bug report be confirmed or denied as a known bug. This is counted as a tech support incident because the user is asking for a custom report of bug status. Most bug reports (well over 95%) do not identify a new bug. Instead, they are errors arising from failure to read the documentation, the FAQ, the Knowledge Base or otherwise are errors of concept or operation. True bugs are very rare.

When a bug report is received it goes into a queue for examination. If it is found upon investigation to be a real bug it gets fixed and eventually appears as a bug fix in an update. Publication of the update is notification of the bug. Updates are published so frequently that it is faster to simply eliminate a bug than to note it on a web page. If you want a custom response as to whether what you think is a bug really is or is not a bug, that requires usage of a tech support incident.

If you do submit a bug report and do not request a reply, Tech Support may contact you as part of their investigation. If so, that does not count as an incident. See the manifold.net web site’s Support pages for additional information on bug reports.

### Updates

Corrections of errata are made through Updates (formerly called Service Packs), which may be downloaded by authorized users from Manifold web sites. Manifold releases from 7x onward will by default advise you when
launched if a new update is available. The latest update automatically includes all fixes and upgrades issued in previous updates.

**How to Avoid Needing Technical Support**

The good news is that answers to most technical support incidents are really simply answers, so simple that users are often embarrassed by how obvious the answer was. That's OK, as it is human nature to sometimes overlook the obvious. The key with technical software like Manifold is to leverage the strong part of human nature, the ability to understand and apply cool new things, by having and applying an adult expectation of the effort required to master those cool new things. That means carefully reading documentation and taking advantage of the many resources on the web and in literature for self-education.

If you read documentation and instructions carefully you will eliminate the need to use technical support in most situations. A very high percentage, over 95%, of tech support incidents are used to ask questions for which the answers may be very easily found in the documentation or installation instructions for Manifold.

An astonishingly high percentage, perhaps 30%, of support incidents involve a letter from a user such as "I have read the ... topic but I don't see the answer to this question" even though the topic includes an obvious answer to that question. Why this happens has been researched, as people usually do not want to waste their time writing to tech support if they have the answer immediately at hand.

Initial results indicate that when some people are in a hurry to find the answer to an important question they speed up their review of documentation to the point that they no longer actually read it. Instead, they skim it quickly. If they are in a big hurry they might skim documentation so quickly they miss even glaringly obvious answers, such as those highlighted with bright red, boldfaced "Important" markers.

The solution to this is to slow down, take your time and read the documentation carefully. If you find yourself uncontrollably skimming documentation and unable to actually read it, try reading out loud. This will force you to slow down and actually read each word.

This is especially important if you are attempting a complex project under a lot of time pressure and have encountered difficulties. When people rush through debugging they are never sure whether or not in their haste they have really checked some possibility, so they go back and check it again and again. This is a terrible waste of time. It's like a person who has lost his or her car keys and in great haste starts going through the pockets of clothing they have worn recently. Some people get into a panic and find themselves going through the same pockets over and over "to be sure."

A better method is not to panic. Slow down. Debug what you are doing systematically. Write down a list of possibilities and check each one carefully and systematically. Consult the documentation on each point and read each relevant topic slowly and very carefully, thinking about what that topic teaches and how it may relate to the task at hand. Keep careful notes on what you have done and the settings you used for various dialogs so that if you ever do decide to contact technical support you can provide an exactly accurate and detailed description of what you have done. Quite often, just the act of writing a detailed account of what you have done will reveal an error.

**Tech Tips**

**Invest in Education** - People who invest in their own education by reading the documentation, discussing issues with other Manifold users in the online community and investing into education via training products or coursework in schools will almost never require technical support. If your time is valuable (whose time is not valuable?) it is worth your while to invest into your own education through training.

Many third parties provide training for Manifold System. The training products sold at [http://www.gisadvisor.com](http://www.gisadvisor.com) have been especially praised by Manifold users in online forums. Many users report that self-paced training products like those sold by GISadvisor.com are an easier path to learning than relying exclusively upon reading documentation.

**Bugs are Very Rare** - Newbies will at times leap to the conclusion that the problem they are experiencing is caused by a bug in Manifold. Anecdotal evidence indicates that the less carefully someone has read the documentation, the more likely they are to leap to such a conclusion. This is almost always a mistake, as bugs in Manifold are very rare and are almost never found by inexperienced users.

It is usually a waste of time and energy to think that whatever problem you are experiencing is caused by a bug in Manifold. That is almost always a psychological drain that will divert your attention from successfully
understanding what is going on. Instead, focus your energy on a detailed understanding of what you would like to do and the Help topics that are relevant. The possibility of a bug is always kept in mind by tech support, but leave it up to tech support to determine if a bug is the source of the problem. This even tech support cannot determine without lots of detailed information so careful attention to what you are doing is necessary in any case.

It's true that just as "even a broken clock is accurate twice a day" on very rare occasion there are new bugs found as a matter of happenstance by beginners. Don't let that rare statistic distract you from starting your analysis by assuming you have missed something in the documentation.

You can, of course, submit a bug report following the guidelines in this topic even if you are not sure that something is a bug or not. Manifold gratefully receives thousands of bug reports knowing full well that almost all of them are not bugs. But all are gratefully received so that even if only one small bug is found by sifting through many reports the system will become better than it is. Collecting and acting upon many bug reports is one reason the system has become so clean that new bugs are very rare, and that's the way everyone wants to keep it. Just don't allow the writing of a bug report to distract you from the more likely possibility of a nuance missed or misunderstood.

In a related note, beginners uncertain of their skills will sometimes panic after reading Internet forum threads discussing "bugs" in Manifold. Almost all "bugs" discussed in Internet forums on Manifold are not bugs. There are a few, of course, that experts discuss; however, most postings that include the word "bug" in the subject line do not discuss bugs but rather end up being a discussion of something the original poster did not understand.

This is such a common effect that the use of the word "bug" in a subject line tends to indicate the poster is either a rank beginner or is easily panicked, as Manifold experts know that new bugs are very rare and whatever problem they are experiencing most likely derives from some fine point that is yet to be learned or has been missed. So they don't normally post using that word in the subject line. Instead, they post using a subject line that describes unidentified behavior as narrowly and neutrally as possible. The last thing an expert wants is to be embarrassed by calling something a "bug" only to learn that it was a simple error on his or her part, so experts tend to avoid that word.

In contrast, beginners will at times launch a thread on some awful "bug" they have discovered when they have simply neglected to learn and apply the software properly. On occasion some people persist in insisting something is a bug even after numerous experts have responded to their post pointing out the problem is not a bug. But still the thread persists using the word "bug" in the subject and that can scare other beginners as yet uncertain of their own mastery of the software. Don't let that bug you!

FAQ

See the Support page on the manifold.net web site for frequently asked questions, Knowledge Base articles discussing what is or is not a tech support incident and other useful information about using technical support.

Why do you charge an incident for even simple questions? There is no way to foretell if a particular email received from a user is a trivial question or a sophisticated question. Each question requires reading by a trained engineer to assure that a sophisticated question that might appear to be a simple question to the untrained eye is properly answered.

For that reason, it costs as much to process an apparently trivial question as it does a carefully thought-out, complex question. In fact, some classes of trivial questions require more effort because users neglect to provide required information or because through lack of experience the user might think something is far simpler than it really is.

Ultimately, since questions must be processed by talented engineers what an incident really covers is the time spent by that engineer together with associated support costs. It doesn't matter if someone is using up that engineer's time with a brilliant question, or if that engineer's time is used up suggesting the customer read a web page, like the Support page, for material that is explicitly discussed on that page.

Why do you ask for so much information? Long experience shows that having all required information at hand is the fastest way of resolving a tech support problem. If Tech Support must repeatedly ask that necessary information be provided, a technical support incident will be charged. For example, tech support questions of the form "I can't open the file - what am I doing wrong?" are a good way to waste incidents. Tech Support must respond by first asking for all information noted as set forth in the Support page on the manifold.net website. Only after that information is provided can Tech Support help the user.
Most tech support questions are very elementary errors or misconceptions. Even the smartest people can miss something very simple, so it is essential in Tech Support that the details be nailed down. Getting the background information is essential to getting the details right.

Some users will not cooperate with Technical Support. The classic case is a user who "knows" his problem is not something dumb he is doing so he refuses to answer questions asked by Tech Support that he feels are demeaning. However, when tech support asks simple questions it is based on long experience that shows even the smartest people will at times make really dumb mistakes that may be obvious to a third party but not to the person struggling with the problem. Such mistakes must be excluded by asking specific questions and getting clear answers, with nothing taken for granted. If a user does not answer all of Tech Support's questions the incident will be closed (and charged to the user).

Why must I have the latest versions of everything and the latest service packs? Manifold uses many Microsoft system facilities to assure perfect compatibility with Windows. Problems within Microsoft code might therefore show themselves as problems encountered when running Manifold. Both Manifold and Microsoft eliminate problems by issuing service packs and by issuing newer releases. Because technical support is a limited resource, it does not make sense to waste it on figuring out and fixing problems that have already been fixed by either Manifold or Microsoft in a service pack or in a new release. Requiring that customers use the latest release of Manifold and install the latest Manifold and Microsoft service packs is the only way to guarantee that problems that have already been fixed are not at issue.

Can I get free tech support by sending a question to Sales? Tech support incidents also apply to questions of a technical support nature that are sent to sales or to any other manifold.net email address. Some users occasionally try to gain additional incidents by sending email that is clearly of a technical support nature to sales or to other manifold.net email addresses. Such emails are forwarded to tech support and will be charged as tech support incidents.

Can I ask more than one question in the same email? When more than one question is asked in the same email, Technical Support will apply an incident to each question in turn until the user runs out of incidents. An incident covers the lowest resolvable usage in Manifold, so any distinctly different questions involve different incidents. For example, a question as to why certain fields are not being transferred may be answered by Technical Support with a note to use Transfer Rules. That's one question with one answer and one incident. A follow-up question by the user asking how to set transfer rules to a particular default setting is a second question that will consume a second incident.

Does tech support assume I've read the documentation? It is unrealistic to assume that someone is attempting to use Manifold without reading the documentation. All tech support interactions proceed based on the assumption that users are in fact reading the documentation. Answers to questions therefore assume that users will use those answers as a guide to reading Help. For example, a technical support answer that cites the need to use Transfer Rules assumes that the user will read the Transfer Rules topic in Help as well as other relevant topics that branch out from that topic.

Technical support answers are therefore not a substitute for reading the documentation. If any information is available in Help, technical support will not repeat it but will assume the user will read such documentation based upon the notes in the technical support reply. Since individual subjects will often branch out from one topic into many different topics, technical support will not enumerate all topics that might be relevant: it is assumed the user will click on hyperlinks and other references to branch out through the documentation as the user sees fit.

What does a tech support incident cover? A technical support incident is any single question asked of technical support. The same incident continues no matter how many emails are exchanged until the single question is answered. "Single questions" mean those that resolve to one function or to the lowest, single, separable usage level of Manifold in a specific task.

For example, the question "How do I use Manifold?" is indeed a single question, but it is not resolvable to one function or to a matter of a single usage level in a specific task. Asking such a general question is a waste of a support incident because it will result in an equally general answer.

General questions will receive general answers, since they usually involve numerous functions or very many lower level usage levels. For example, a general question such as "How can I link to external database tables" will receive a general answer pointing out various Manifold Help topics to read, which will cost one incident. The thread does not continue into subsequent specific questions, such as, perhaps, how to operate an ODBC dialog, how to configure one's SQL Server database system and so on.

It is possible, of course, if one purchases enough support incidents to use tech support as a replacement for reading the documentation. However, such usage is very inefficient and expensive. A better idea is to get some general education in the subject matter, to read the documentation or to hire a consultant with the required expertise.
Excluded from standard support are questions related to any form of scripting, any other programming, any customization (through any of the methods that might be used to customize Manifold, such as custom datum projection grids, geocoding data extensions, or the various other customization features), SQL, Enterprise Edition features, Manifold IMS and any question involving a Runtime license. These topics require development level support incidents.

Excluded also are questions that are of a consulting or educational nature or that describe the operation of Windows in general or products other than Manifold System. Only supported are the specific functions and capabilities of Manifold.

Examples of supported questions:

- "How do I import a projected .shp file?"
- "Can Manifold import .ijk format for surfaces?"
- "How do I rotate a label?"
- "How do I link a table into a project?"
- "If I do not have enough free space in my Windows TEMP folder, will that be a problem?"

Examples of unsupported questions:

- "What is the difference between .shp files and .mid/.mif files?"
- "How do I write a script to import .ijk format for surfaces?"
- "How are labels stored in AutoCAD?"
- "How should I configure SQL Server with the right permissions for me to link a table into my project?"
- "How do I change the location of my Windows TEMP folder?"
- "What is wrong with my XML customization?"
- "Microsoft IIS won’t launch on my machine. What am I doing wrong?"

Note also that Manifold Technical Support provides technical support on Manifold products, not on Microsoft facilities or on other products. Manifold can work with many other software products. For example, one can create a linked drawing in Manifold that draws its contents from a SQL Server table. However, support on Manifold is limited to Manifold only.

For example, if you are having difficulty creating a linked drawing because your SQL Server system or Windows security is configured in a way that prevents your user login from accessing the desired SQL Server table then Manifold tech support will not undertake what is, in effect, a consulting project to help you re-configure SQL Server or diagnose the improper administrative setup of Windows.

Manifold exposes a variety of interfaces that may be used by third parties to create add-ins or other modules that work with Manifold. Manifold Technical Support does not support such third party code. For example, Manifold Technical Support does not support any add-ins from third parties, image server modules from third parties, geocoding modules from third parties or any programs utilizing Manifold runtimes.

Important: Does manifold.net support the DBMS products provided for free by DBMS vendors? No. Manifold.net does not provide any support whatsoever for installation, administration, management, configuration or use of the DBMS products provided for free by DBMS vendors.

Free packages are discussed for the convenience of users only: They are not supported by manifold.net and they are not supported by the DBMS vendors. If you want a supported DBMS installation, you must purchase a supported DBMS product from one of the DBMS vendors or you must purchase support products, if such are available, for the free DBMS product you are using from the vendor of that DBMS product. Manifold.net does not sell support products for the free DBMS products provided by DBMS vendors.

The DBMS products provided for free are each supported by a vast industry of educational and training materials. Hundreds of books, for example, have been written on SQL Server alone, ranging from texts aimed at “dummies” to those covering the most sophisticated uses imaginable. Users intending to take advantage of these DBMS products should acquire and utilize the appropriate educational materials.
Important: Do not waste technical support tokens by sending in tokens for questions about the free DBMS products. Doing so will simply waste a token to no good purpose.

Developer level support incidents may be used for questions about Enterprise Edition features used with these database products, but only for the actual Enterprise dialog. For example, a question about connecting with Enterprise Edition via Database Console to an Oracle data source will result in an answer limited to the use of the Manifold dialog, such as explaining the purpose of the Server, User Name and Password boxes. Support does not extend to explaining how you can configure users in the DBMS, how to determine what server name is being used or should be used or how to determine whether a given user has connectivity to a given data source.

What do developer technical support incidents cover? As noted in the above comments, developer level incidents cover all aspects of Manifold System, including scripting, any other programming, any customization (through any of the methods that might be used to customize Manifold, such as custom datum projection grids, geocoding data extensions, or the various other customization features), SQL, Enterprise Edition features, Manifold IMS and questions involving a Runtime license.

Excluded also are questions that are of a consulting or educational nature or that describe the operation of Windows in general or products other than Manifold System. Only supported are the specific functions and capabilities of Manifold.

Note that as with standard technical support incidents, developer incidents cover a single question that relates to one function or to the lowest separable usage level of Manifold in a specific task. General questions will get a general response. Exploration of sub-parts of that general response or more specific questions for more detailed "how to" guidance on sub-parts will require additional developer incidents.

Investigation of development problems will often involve more than one technical issue, with more than one error in usage or the need for guidance in more than one area. Sometimes a sequence of problems must be identified and solved before an overall problem can be identified and cured. That will require a sequence of developer support incidents. Each problem solved, even if only for diagnostic purposes to set aside a possible cause, uses up a developer support incident.

Developers are strongly advised not to waste support incidents on simple matters that may be found in the user manual, in examples published on the manifold.net web site or that may be found in a careful search of past threads on Internet forums for Manifold System, such as forum.manifold.net. It is also a good idea to first achieve expertise in the development environment being used so that developer incidents are not wasted on matters arising from weak understanding of the target development environment.

Developers will get maximum value when using developer support incidents if they have strong skills in the development methodology that they choose. For example, if a developer does not have strong Visual Studio expertise it is easy to waste developer incidents on questions arising from generic operation of Visual Studio that have nothing specific to do with Manifold. In such cases it would be more effective to develop greater skill in Visual Studio programming using general Microsoft educational resources such as MSDN and to save developer support incidents for issues specific to Manifold. Note that Manifold developer incidents are not a support path for non-Manifold tools such as Visual Studio.

Developer incidents are not a consultation service. Although general questions of how to approach a particular task are appropriate for a developer incident (and will receive a very general answer in reply) a developer incident cannot be used as a replacement for the services of an expert consultant. Many experts provide Manifold consulting. Recruit such experts with postings on forum.manifold.net if you need a consultant.

For example, developer level incidents may help zero in on an approach to a new project. Someone might ask, "I have the following three approaches I might take and here are the pros and cons of each as I understand them. Which do you recommend?" Although the answer to such questions will be brief and general, a pointer in the right direction from a manifold.net expert will often be very valuable for the guidance it gives. If taking this approach, keep in mind that such guidance will be brief and general. Further discussion, clarifications, assistance in writing code and so on will all consume additional incidents and will not be cost-effective compared to hiring a consultant to assist in the project.

Developer incidents are not a code-writing service. At times an answer from tech support might include snippets of code to illustrate a particular point (at the option of the tech support engineer writing the response), but this only occurs in very specific matters like the functioning of a particular object in a highly specific context. Anything other than such very specific snippets consisting of one or two lines of code should not be expected. Requesting that tech support send you a code sample for a particular task will waste an incident.

The bottom line for developer incidents is that they are a good way for people who have a mature and expert understanding of Manifold interactively and also a thorough understanding of their development environment to...
ask specific questions about specific features. For example, when you are deep inside a programming project and have hit a specific roadblock involving a particular object, a developer level incident can work magic to part the mists of confusion and get past that roadblock.

**Why don't you provide free technical support?** Regrettably, there is no such thing as "free" technical support any more than there are perpetual motion machines. If there was such a thing as free technical support we would be the first to take advantage of it!

In real life it is a very expensive proposition to keep expert engineers at hand to answer technical questions. Salaries must be paid, equipment must be provided and facilities and other resources must be funded. All that money comes from somewhere, and in a commercial company that means that every dollar for any cost associated with providing technical support comes from money paid by customers. In one way or another customers pay for technical support as that is the only source of revenue in commercial companies.

The money taken from customers to pay for technical support can come from some percentage of money paid for the product or it can come from extra fees that are charged those who use technical support or it can come from some combination of the two. But no matter what it all comes from the customer. The only question is what is the most fair and efficient way to charge customers for technical support.

Those companies who say they provide "free" technical support for a product are really telling a lie, because as we have seen above the customer is paying for that technical support in some way. In the case of companies who mean by "free" that they do not charge extra for specific uses of technical support, what is really going on is that some portion of the money all customers pay to buy the product is set aside to pay for the costs of running the technical support function. If that money were not used for technical support, the product could have been sold to customers for a lower price. The technical support in such cases is not free; it is just hidden as a cost within an inflated product price.

Since most users do not use technical support while a small percentage of users will abuse "free" technical support, what ends up happening in such situations is that everyone must pay a higher price for a product just so that some people can be subsidized with extra services paid for by other people. We don't think that is fair.

**Manifold.net** does not burden the price of Manifold System with hidden costs because we believe that doing so would be an unfair transfer of costs from some customers to other customers. If we were to provide "free" support on Manifold System, we would have to increase the cost of the product to pay for that support.

If we were to increase the cost of the product to all users to provide "free" technical support for everyone, we would in effect be charging those users who do not need technical support to provide a special service to those who do. It is especially unfair to all customers if that special service is provided (at great cost) to those who won't read documentation or invest in their own educations because they know that others will pay for those lapses.

Even setting aside egregious abuses, what one person thinks is a fair use might not be reckoned fair by those who are called upon to pay for it. For example, a consultant who is under intense time pressure to complete a task and who does not have time to read documentation thoroughly may consider that situation "business as usual" while another user may feel that part of budgeting a project is setting aside enough time to get educated in the use of required tools. The only way to respect what everyone considers fair in such circumstances is to let everyone pay for the resources they use.

We feel the fairest system is to not charge for technical support through hidden transfers but to directly and openly charge people for whatever support they choose to use. In that way, those people who prefer to use free resources like self-education, documentation and Internet forums to answer questions are not forced to pay for those who don't, and those people who find it more expedient and efficient to pay for technical support can do so.

The reader should understand that the above comments on people who abuse "free" technical support are not intended to be disrespectful to those who use technical support in the normal way. It can make sense from time to time even for the most diligent, the best managed, the best educated and the very brightest people to seek help from technical support. It's just that we think the fairest way of running matters is that when someone does decide to use technical support (for whatever reason) that person should pay their own costs of doing so.

**See Also**

**Contacting manifold.net**
**Help - Manifold on the Web**
**Troubleshooting**
Activation and Licensing

Activation Keys and Serial Numbers

Manifold is licensed, not sold, and your Manifold license is not without limits. It only allows you to use Manifold within the limited terms and conditions of the license. Important limitations on usage will apply, so be certain to read this topic carefully to make sure you can get maximum use of your Manifold license.

This topic talks about the nuts and bolts of activating a Manifold installation. You should read this topic before activating Manifold.

All Manifold licenses are copy protected using one of two mechanisms:

- Authorization using a Serial number and Activation keys that discourages software piracy. Manifold licenses acquired in small quantities or acquired for use on portable devices use this mechanism. Continue reading this topic to learn about the serial number and Activation key process.
- Authorization of floating licenses using a License Server. Manifold licenses acquired in larger quantities will frequently use this mechanism, which requires licensing of the Manifold System License Server product. See the Manifold System License Server topic for discussion of License Servers.

The rest of this topic assumes you are using a serial number to activate Manifold on a single computer system.

For Manifold x64 Licensees: If you have an x64 Manifold license and will be operating Manifold in either a 32-bit or a 64-bit Windows system, please read the 32-bit and 64-bit Manifold Editions topic after carefully reading this topic.

Activation Summary

Login as Administrator. Use the actual Administrator account and not some other account believed to have administrator privileges. Vista users: please read the advice in the Vista Admin Login page on the Support page of the manifold.net web site to see how to use the actual Administrator account, which might be concealed by Vista editions, such as Home edition.

Install Manifold and then launch Manifold. Check your computer's date and time to make sure it is reasonably accurate (the correct year, month and day). In the Activation dialog that pops open, provide the serial number emailed to you for the Manifold edition you have licensed. Copy the serial number from the original serial number email message and paste it into the dialog.

For the first 30 days after your serial number was issued, you can run on the serial number alone, and you can install that serial number on as many computers as you like. There is no limitation of Manifold functions running on serial number alone: all features for whatever product you have licensed are fully functional and there is no reason to activate the product.

After 30 days you will have to get an Activation key as well. When the Activation dialog launches, enter your serial number and then press the Get Activation Key via the Web button and then press Accept. Do not press the Get Activation Key via the Web button unless you really intend to get an Activation Key. Once that button is pressed the operation cannot be cancelled. Getting an Activation key terminates any return privileges you may have.

If you have licensed Business Tools, Geocoding Tools or Surface Tools extensions: launch Manifold and choose Help - Activate Extension. Enter the serial number for the extension. Get an Activation key if it is more than 30 days since the serial number was issued. Press Accept. Do this for each extension that you have.

If you have licensed Universal Edition or Ultimate Edition, all three extensions get turned on automatically when you activate your main Manifold System serial number. No need to use the Help - Activate Extension dialog with Universal Edition or Ultimate Edition.

Important: Manifold is licensed for use on one machine at a time. You get five Activation keys per serial number to allow re-installs if necessary, not to allow simultaneous use on five machines. Don't waste Activation keys. Read this entire topic carefully for tips on getting the most out of your license. Installing the same serial number
on multiple machines and using that same license on multiple machines at the same time is not allowed by your license and may result in your license being permanently revoked.

Caution: If you do not have Administrator privileges the Activation dialog will accept the Activation key and will allow you to run for that session, but the installation will not be permanently accomplished. This feature enables emergency activation of Manifold on a temporary basis if an Administrator is not available. If you do not remember to use the Activation key with an Administrator login during the 72 hour life of the Activation key you will have wasted that Activation key. If you do not have Administrator privileges when the Activation dialog is launched, Manifold will warn you that any activation will be good only for that session. Don’t make the mistake of ignoring such warnings!

Troubleshooting / Reminder for Experienced Users

This checklist is here as a quick reminder for people who are re-installing Manifold and have not read this topic for a while. New users should read this entire topic. In case of any difficulty with activation, consider the following:

- **Check the serial number**: Check the serial number at the Status Page, which is a link on the manifold.net Support page (http://store.manifold.net/online/status.aspx - check the Support page if this link has moved in the future). The Status Page will tell you whether a serial number is valid, what product it licenses, how many activations it has available, and whether it has been revoked. The most common reason a serial number has been revoked is because it has been traded in on some upgrade, like a Professional serial number being upgraded to Enterprise or a 32-bit Manifold edition upgraded to x64. In that case you should use the new serial number provided by the upgrade.

- **Typographical errors**: Copy and paste serial numbers from the original serial number email. Do not change the serial number in any way.

- **Wrong serial number**: Make sure your serial number is for the product you are trying to activate. You can't activate a Release 8.00 license with a Release 7x serial number. You can't activate a Manifold System installation using the serial number for an extension. You can't activate an x64 license with a 32-bit serial number.

- **Wrong system date**: Your system date must be accurate. Manifold will not accept serial numbers or Activation keys if the system date is not reasonably accurate.

- **Wrong machine**: A serial number and Activation key pair are good for one machine and one machine only. You can't use a serial number and Activation key on a different machine.

- **No Activation key**: A serial number will work by itself for 30 days after the date of issue. After that, you must get an Activation key as well.

- **Expired Activation key**: An Activation key is good for one machine for 72 hours only. When you get an Activation key, use it within 72 hours. If you ever need to re-install Manifold on the same machine, you cannot re-use an Activation key obtained more than 72 hours previously. You'll need to get a new Activation key.

- **Same key every 24 hours**: You must wait at least 24 hours before getting a different Activation key for the same serial number. This is to discourage piracy and to save people from accidentally using up their Activation keys.


When installing or activating Manifold, you must have a standard Windows installation in operation. Some registry "protectors," third party uninstall programs, network supervisors and other "value added" supervisory programs running on top of Windows may interfere with the installation and/or activation process. Temporarily turn off third party virus programs as well. Manifold software and these instructions assume you are running an unmodified Windows installation as shipped by Microsoft.

Keep in mind standalone Manifold licenses are licensed for standard Windows installations on physical machines. Standalone licenses are not licensed for virtual machines (use License Server with appropriate numbers of floating licenses). Likewise, Manifold is not licensed for time sharing nor can a single Manifold license be used on more than one machine at the same time.

**Assistance**

Assistance may be available for the original licensee of a Manifold license - see the Support pages in the manifold.net web site for any current free or paid technical support offerings for activation.

There are usually two ways to get assistance with serial number and/or activation questions:
• **Free support** using the procedures published in the Activation Support page in the manifold.net website.

• **Paid support** using a series of guided questions and answers with tech support, with a support token required for each such question and answer session.

All serial number and activation questions are handled by email questions sent to tech@manifold.net and must follow the procedures published in the Activation Support page in the manifold.net website. Do not contact manifold.net by letter, fax or telephone if you need assistance with any questions or issues involving serial numbers, activation or software installation. Technical support processes email inquiries from 9 to 5 PST on US business days and responds to activation questions before any other questions.

You do not need a technical support token for activation questions if you follow the procedure given in the Activation Support page. If you do not wish to follow that procedure you may use paid support to get assistance if you are the original licensee. If you contact tech support outside of the required procedure, you may be charged a tech support token with any tokens you have available being automatically consumed for such out-of-process contacts.

Please see the Activation Support page for the current procedure to follow. As a general guide for the information and procedure that web page will likely require, here is a list of information that will probably be required in your initial email message:

• Use a subject line that includes the words **Manifold** and **activation**. This will assure your message gets through any spam filters and receives maximum priority for a reply. Activation questions are answered before all other questions.

• Do **not** use any attachments. All information must be provided in ordinary text form within the body of your email message.

• Do **not** use HTML mail. Send ordinary text mail to be sure all information gets through.

• Do **not** cc any third parties. Serial number, activation and installation questions are confidential.

• At the bottom of your email you **must** include the full text of the original serial number email that provided your serial number for this license. Copy and paste this text into the body of your email. Do not attach it as an attachment. Possession of this serial number email is proof you have licensing rights to this license. If you do not have the original serial number email message you cannot get any assistance. If you do not have that serial number email, you might be able to recover it using a Key Recovery Service if available on the Manifold Online Store.

• Specify the full name of the installation file from which you installed Manifold.

• Confirm you have logged in as the Administrator account. This is especially important in Windows Vista.

• Specify the exact version of Windows you are using.

• Give the system date and time as reported by your system at the time you try activation.

• Include in your email the exact serial number, Activation key and System ID you are using. Use copy and paste to insert these into email message to avoid typographic errors.

• Provide any error messages you receive or an exact description of the problem if you received no error messages.

If you don't want to follow the procedure in the Activation Support page that is OK: if you prefer, tech support can guide you by asking you a sequence of questions, one at a time, to elicit all required information. That will require a tech support token for each such question and answer. See the Contact Tech Support page on the manifold.net website for information on purchasing and using tokens if you would prefer to use the guided process.

**Tech Tip:** The fastest way to solve any activation problem is to read and apply the instructions in this topic, followed up by a reading of the relevant support web pages on the manifold.net website such as the Activation Guide and the Activation Troubleshooting pages. Even though tech support responds very rapidly to activation questions it is faster to read even a long topic carefully since that will usually solve the problem right away.

**Your Serial Number**

Your Manifold **serial number** identifies your license to use a specified Manifold System edition or a Manifold System extension such as Business Tools, Geocoding Data or Surface Tools. A serial number specifies a product license, it is not locked or tied in any way to one person or to one machine. It is not locked or tied in any way to any product download. For that reason, it is critically important to keep your serial number secret. Don't think that any registration or activation procedure locks a serial number to you: it's always available to anyone who has it, so don't give a serial number to anyone you don't want to be able to use it.
If someone else learns your serial number, they can use it to steal your license rights from you. They can download a product installation, and use the serial number to install Manifold on other machines. When they start using the same license on different machines at the same time you might be using Manifold, it could be that your license will be revoked. Further, your Manifold license requires you to keep your serial number secret. If you disclose your serial number to third parties, you have violated your Manifold license. Each serial number is licensed to be used on one machine by one person, and a copy-protection mechanism using that serial number aims to prevent software piracy.

The purpose of a serial number is to identify the product it licenses. It's like a price tag or an SKU tag on an item in a store. A serial number works only for the product for which it was issued. For example, a Professional edition serial number cannot be used to turn on Enterprise edition. For example, a 32-bit Personal edition serial number won't turn on a 64-bit Personal x64 installation. The email in which your serial number is delivered to you makes it clear what product that serial number is for. Keep track of your serial numbers - that is easy to do if you are careful not to lose the original serial number email.

For permanent installation and continued use, your license requires you to activate your Manifold license. Activation requires fetching an Activation Key that matches the serial number. This may be done through the web in a fast and easy manner or it may be done manually if desired. Once a Manifold license is activated it will be permanently installed on one computer system and will run without time limit on that computer.

Important: Each serial number allows five activations. The purpose of those five activations is allow re-installation in case of hardware difficulties or upgrades. The purpose is not to license usage of one license on five different machines, although if someone tries that as a practical matter the limit on activations in a relaxed way limits the number of different people who can use the same serial number to install Manifold on many different machines. If you use the same serial number to try to run one Manifold license on more than one machine at the same time, Manifold will automatically revoke that serial number and you will lose your licensing rights.

The Manifold serial number and activation system is very simple; however, it is the nature of comprehensive documentation to include explicit cautions against errors encountered by some users. What may seem obvious to the reader may not be so obvious to someone else, so we have erred on the side of too much detail rather than too little. We apologize in advance for the resultant length of text that everyone thus must plow through.

Activation Keys and Serial Numbers Overview (Short)

This section provides a short Overview section of paragraphs that cover the most important points. Following the overview section is a horribly long Details section that goes into greater detail about activation. It's best to read both sections.

Administrator Permissions Required

You must have Administrator permissions in Windows 7, Windows Vista, Windows Server 2008, Windows Server 2003, Windows XP or Windows 2000 when doing any Manifold installation or activation. The easiest way to assure you have Administrator permissions is to login using the built-in Administrator login. If using XP Home in default configuration any user login with "Computer administrator" rights is OK.

Windows Vista users should login using the built-in Administrator login to avoid problems caused by logins that are thought to have Administrator permissions but do not actually have them. Vista users: please read the advice in the Vista Admin Login page on the Support page of the manifold.net web site to see how to use the actual Administrator account, which might be concealed by some Vista editions such as Home edition.

Launching Manifold for the first time or attempting to activate Manifold or a Manifold extension will display a warning message if the user account being used does not have sufficient permissions to complete the activation process.

Shut off all other programs when doing Manifold installation or activation, including any antivirus programs, network snoopers, uninstall utilities, registry protectors, Windows modifications, virtual machine middleware and other weird stuff.

If you see an Authentication Error message box pop open during any part of the activation process, something is interfering with that process. The usual problem is either failure to have Administrator permissions or failure to shut off some background program, such as an antivirus program, that interferes with activation.

Manifold is Copy-Protected
Manifold is copy-protected against software piracy, using a combination of technical means and incentives drawn from human nature to discourage people from stealing licenses. Most people who use GIS don't want to steal software, so Manifold can use a more relaxed approach to copy protection.

The Manifold copy-protection mechanism relies in part on technical mechanisms and also in part on human nature. The technical part happens when a serial number gets an allowed installation used up during the activation process. The human nature part happens when the activation process is policed by you, because you only get five activations for each license. If those five activations are used to try to cheat the license, you may find yourself without the ability to re-install Manifold when you need to because of a hardware failure or machine upgrade.

The purpose of providing five activations for each license is not to enable you to install one license on five different machines for five different users. One license can only be installed on one machine for use by one user at a time. The purpose of providing five activations is to provide the ability to re-install Manifold if you encounter a hardware failure, upgrade a machine or otherwise need to re-install Manifold for use on a different machine.

If you run out of activations, you'll probably want to buy a fresh license or more activations for an additional fee. This policy is strictly enforced, because otherwise software pirates would have no incentive not to rip off the system.

The Activation Process

For 30 days after a serial number has been issued, that serial number can be used to enable launch of Manifold on as many machines as you or whoever has the serial number wants. This initial 30 day period allows users to resolve any difficulties with software or hardware configuration on the machine they decide to use for the license. There are no limits at all: the software runs with all features enabled and no limitations. After 30 days, the Manifold installation can be activated to enable Manifold to run forever on that machine.

Note that activation cannot eliminate software piracy: that's done by detecting when the same serial number is used on more than one machine at the same time. Activation is far too relaxed a control to eliminate software piracy, but it does provide a reasonable way to discourage outright piracy in countries where software piracy is illegal.

Activation can be accomplished automatically if the machine is connected to Internet. When the Activation dialog launches, enter your serial number and then press the Get Activation Key via the Web button and then press Accept. If the machine is not connected to Internet, activation can be accomplished manually by using a different machine that is connected to the Internet.

If a Manifold installation is not activated within 30 days of the serial number being issued, Manifold will no longer launch on that machine. The serial number will still identify a Manifold license, but after 30 days it can only launch Manifold if activation is also done.

Note: The preliminary installation period before activation is required is 30 days from the date the serial number was issued. It is not 30 days from the date you receive Manifold and not 30 days from the date you install Manifold.

Activations are Limited

You only get five activations per serial number, so don't waste activations. Use the first 30 days to sort out any machine problems or other software installation problems and don't activate your Manifold installation until you are sure you want to install Manifold on a particular machine.

Plan your installations so that activations are not wasted. If you know you are going to install a larger hard disk into a particular machine, don't activate Manifold on that machine a week before you will install the larger hard disk. That will waste an activation key on only a week's functioning and then when you install the larger hard disk you'll need to use up another activation. It would be smarter to install the new hard disk first and only after do a Manifold activation.

Install anti-virus software and use prudent Windows administration policies so your machine does not get trashed by viruses every month and thus forcing an overly-frequent need to re-install Windows. If you re-install Windows, you'll use up another activation key.
Five activations are plenty if you install Manifold only on one machine as allowed by the license and if you follow prudent machine and Windows administration policies. Manifold can run for years given five activations. In fact, given Manifold's traditional upgrade policy that provides fresh activations, the provision of free activation keys over time and the availability of additional activation keys at a low cost it is really difficult to run out of activations if you take advantage of low cost or free opportunities, even if your machines die from time to time.

Once installed a Manifold license will run as long as that computer continues to operate in sufficiently unmodified form to not require re-activation. The five activations provided for a serial number enable re-installation and re-activation within a reasonable period of time; however, re-activation of old or obsolete licenses may not continue to be supported once future versions of Manifold are released. Although manifold.net may voluntarily support use of activation keys for outdated versions of Manifold for some reasonable period of time (such as three years) licensees should not expect such support.

**Purchasing Additional Activations**

The least expensive way of assuring a regular re-charge of activations, should such be needed, is to take advantage of low-cost upgrade offers whenever a new Manifold release is announced. Every few years, Manifold System has historically been upgraded to a new release and manifold.net has traditionally offered the new release to licensees of the previous release for a nominal charge for a limited period of time.

The cheapest way to get additional activations is to upgrade to the new release during the initial, low-cost offer. For example, during the first 30 days of the availability of 8.00, 7x licensees could have upgraded to 8.00 for only $50. The low upgrade fee delivered a new serial number complete with five fresh activations.

In addition, Manifold normally provides an additional activation key every year for free and provides absurdly low cost options to purchase additional activation keys in amounts that are reasonable for individual users who are observing the license. This makes it easy and inexpensive or free for law abiding users to continue to operate Manifold for years despite machine failures without encouraging software pirates to cheat the license on an industrial scale.

If you take advantage of free activations, inexpensive additional activations or upgrades to new releases, you can get a new serial number and five fresh activations for either free or for a very low cost on a regular basis for fresh activations. If you are using Manifold on one machine as permitted by the license, it is virtually impossible in ordinary usage to burn up five activations in only 18 months.

On the other hand, if you don't take advantage of upgrade offers, if you ignore free activations, if you don't take advantage of very inexpensive additional activations or if you have special needs (such as usage in harsh environments that regularly destroys machines), then your costs of getting activations could be higher, possibly much higher if you need to buy new licenses on a regular basis. If taking advantage of many opportunities for low cost or free activations does not provide sufficient activations, it could be that you will have to buy additional Manifold licenses to get the activations you need. For example, someone who operates Manifold on a portable device in an extraordinarily harsh environment such as oil well exploration or in a war zone and finds themselves having to replace their computer every month should probably budget for additional Manifold licenses as well.

**Note:** If you are a member of an organization that is operating a Manifold System License Server you normally will not have to plan for activations because you can take advantage of floating licenses administered by the License Server. This can be a highly cost effective route for some organizations.

**Activation via the Web**

When Manifold has yet to be activated, it will raise the Activation dialog every time Manifold launches. This dialog can also be launched from within Manifold by using the Help - Activate command.

To activate Manifold automatically via Internet:

- Launch Manifold and in the Activation dialog provide your serial number in the Serial number box and then press the Get Activation Key via the Web button.
- When the Activation key appears, press the Accept button. That's it!

Activating Manifold automatically via the web will connect to the Manifold activation service, will automatically send the activation server your serial number and the system ID and will fetch an Activation key, decreasing by one the number of activations allowed for that serial number in the central Manifold serial number registry.
If the serial number has already been used to accomplish five activations or if the serial number has been revoked for any reason, the attempted activation via the web will fail.

If you have previously provided a serial number to allow Manifold to run for a limited period before activation, when the Activation dialog opens it will show a masked version of the serial number, with X characters concealing the trailing portion of the serial number. You cannot use this masked version for activation. You must paste the original, complete serial number into the dialog before getting an Activation key over the web.

**Note:** When we press Get Activation Key via the Web Manifold must be able to connect to the Manifold activation web service via the web. If Manifold cannot connect to the Manifold activation web service a dialog will pop open allowing editing of proxy server configuration in case a proxy server is blocking outbound web access.

**Manual Activation**

If you prefer to activate Manifold manually you can do so as follows:

- Launch Manifold System and with the Activation dialog open, copy the System ID string. A System ID is a long alphanumeric sequence that comprises a “fingerprint” for a specific activation request.
- Visit the Manifold activation web site at [http://www.manifold.net/activation](http://www.manifold.net/activation) and paste the System ID into the System ID box. Paste the serial number provided to you for your license into the Serial number box.
- Press the Create Activation Key button and carefully write down or copy the resulting Activation key. This Activation key is good only for 72 hours from the time it is issued. After that, it is useless.
- Back in the Manifold System Activation dialog, enter your serial number and the Activation key obtained from the Manifold web site and press Accept.

Because serial numbers, System IDs and Activation keys are all long sequences of numbers and letters, it doesn't make sense to do manual activation if there is any possibility of doing activation automatically via Internet.

Note that the words used to name the various parts of the activation process are not intended to be accurate as a matter of science: they are only convenient names to refer to the process. For example, the System ID doesn't actually get computed from the hardware - it cannot be "gamed" by making an exact duplication of the hardware. It's just a way of identifying a specific activation request so that people won't burn up many Activation keys by accident.

**Wait 24 Hours between Getting Activation Keys**

**Important:** Using the same serial number within any 24 hour span to fetch an Activation key, even if the System ID is different, will result in the same Activation key. This is both a safety measure and an anti-piracy measure. It is a safety measure so that people who mindlessly push buttons multiple times on web sites or dialogs don't use up all their Activation keys in a matter of seconds, and it is an anti-piracy measure to discourage cryptographic attacks and attempted installation on multiple machines.

**Important:** The above means that you should wait at least 24 hours after getting an Activation key for your desktop machine before getting an Activation key using the same serial number to activate a Manifold license installed on your portable computer. If you don't wait 24 hours, you'll get the same Activation key back even though you used a different System ID for the portable computer. The resultant Activation key, therefore, won't work. This problem is easy to avoid: simply wait at least 24 hours after using a serial number to activate Manifold on your desktop machine before using the same serial number to activate Manifold on your portable computer.

**Checking Status of Serial Numbers**

To check the status of any particular serial number, visit the Support page on the [www.manifold.net](http://www.manifold.net) web site to find a link to the Serial Number Status page.

The status page may be used to find out if a particular serial number is a 32-bit or 64-bit serial number, whether it is still active or has been revoked (as might occur if it has been traded in for an upgrade), what product it authorizes and how many Activation keys are available. This same page may also be used to check the status of a technical support token.
A reminder: Any use of a serial number, such as checking status via the Serial Number Status page, should be done using Copy and Paste from the original serial number email. Do not use the "masked" version of the serial number that may be displayed in the Help - About dialog which ends in a series of X characters. The masked version displayed in Help - About has had the final characters altered with a series of X characters so that someone who has physical access to your computer cannot steal your Manifold serial number. The masked version displays enough of the serial number so you can determine for your internal record keeping which serial number you used on a particular machine, but not enough of the serial number for someone to be able to steal it and use it to obtain Activation keys or other wise use it.

Use Email for Activation Questions

If you have difficulty with your Manifold serial number or with the activation process, contact manifold.net by email. Follow the procedures set forth at the beginning of this topic. To discourage people who steal serial numbers and then try to get manifold.net to help them put those serial numbers to use, all serial number and activation key questions are dealt with by email only.

Runtime Licenses

If you ordered a runtime license for Manifold, then the serial number will configure operation in runtime mode only.

In runtime mode, Manifold does not provide a GUI (Graphical User Interface) for interactive use. In runtime mode, Manifold can only function as Manifold IMS or under programmatic control of another application. Attempting to launch a runtime license in interactive mode will display an error message.

Runtime editions of Manifold System make it possible for technical experts to license Manifold at reduced cost for IMS use only or for embedded use of Manifold System within programs they create. Runtime editions may be licensed at a discount from regular pricing charged for non-runtime editions.

Note: Runtime licenses are for tech professionals who have programming expertise, substantial web mastering skills and Manifold skills. Any tech support questions involving runtime licenses require a developer level technical support incident. This includes questions about activation, which normally do not require a tech support incident when non-runtime licenses are involved.

See the Runtime Licenses topic for additional information and tips when working with runtime licenses.

Serial Numbers Determine the Edition

Your serial number turns on whatever product level you have licensed. For example, a Professional Edition serial number will configure Manifold with all Professional Edition features. An Enterprise Edition serial number will configure Manifold with all Enterprise features.

Optional Extensions

Serial numbers and Activation keys are also used to "turn on" optional extensions to Manifold, either as separate installations or as part of a product bundle such as Universal Edition. For example, the Business Tools, Geocoding Tools and Surface Tools extensions are turned on using serial numbers that enable use of these extensions. If you acquire Business Tools, Geocoding Tools or Surface Tools as a separate product you will receive a serial number for each extension you licensed.

If you license Universal Edition, then one serial number will turn on all Universal features, including Business Tools, Geocoding Tools and Surface Tools.

Extensions to Manifold System use the same serial number and activation process used by the main Manifold System license. A serial number for an extension will allow use of that extension without activation for 30 days from the date that serial number was issued. After 30 days, activation will be required for continuing use of the extension.

Important Notes
Installing and activating Manifold consists of several steps:

- **Serial numbers and Activation keys are confidential information.** Keep them secret to prevent other people from stealing your license rights from you. A serial number is like a PIN code to a bank account. *Keep it secret.* There is nothing about a serial number that locks it or ties it to you, that ties it to any particular machine or that locks it or ties it to some download. If someone else learns the serial number, they can use it to steal your license rights from you, for example, by using up all of your Activation keys or by trading in your license for an upgrade. Further, your Manifold license requires you to keep your serial number secret. If you disclose your serial number to third parties, you have violated your Manifold license.

- **Do not lose your serial number.** You will need your serial number if you must ever re-install Manifold. Save a copy of the original serial number sent to you. If you lose your serial number, it may not be possible for manifold.net to provide it to you again. If it is possible at all (unlikely - don't bet on it), a fee for recovering the serial number will be charged. We're not kidding; think of that serial number as a prepaid cash card that anyone can use and if it is lost cannot be recovered.

- **Activations are limited.** Each serial number is a license for only five activations. These five activations allow you to re-install Manifold up to five times for a single license. They are **not** a license to install Manifold on five machines for use by five users but rather give you five chances to re-install Manifold on the same machine or for the same user should a disk failure or hardware or software upgrades require re-installation of Manifold. If you need more than five activations, you may have to pay to buy additional activations or to buy a new license.


- **When acquiring multiple licenses for Manifold or Manifold options,** take care to accurately **keep track of your serial numbers.** Manifold **does not** keep track of these for you. For example, if you acquire Universal Edition and also a Universal Runtime license do not confuse the two serial numbers. Keep a record (such as the original email providing the serial numbers) of which serial number corresponds to Universal Edition and which serial number corresponds to the Universal Runtime license.

- **If you don't want to keep track of many serial numbers,** acquire and use a Manifold System License Server to administer "floating" licenses that are authenticated from a single server.

**Activation Keys and Serial Numbers Details (Long!)**

The rest of this topic covers all aspects of the serial number and activation process in greater detail. It will appeal to the detail-oriented reader and will be cursed by everyone else.

**Installation and Activation Workflow**

Installing and activating Manifold consists of several steps:

- **Install Manifold from a downloaded installation file using the installation instructions that are part of the download.** You must have logged in as Administrator in Windows Vista and your login must have Administrator privileges in Windows 7, Windows Vista, Windows Server 2008, Windows Server 2003, Windows XP or Windows 2000 to install Manifold System.

- **Launch Manifold.** In the Activation dialog that pops up, provide the Manifold System serial number that was sent to you when you acquired your Manifold license. The serial number alone will enable Manifold to function for 30 days from the date the serial number was issued. You can install it on however many machines you like.

- **If you have licensed any extensions, such as Business Tools, Geocoding Tools or Surface Tools,** launching Manifold open the Help - Activate Extension dialog. Provide the serial number sent to you with your Business Tools, Geocoding Tools or Surface Tools license. Just like your main Manifold System serial number, the serial number alone will enable the extension to function for 30 days from the date the serial number was issued. **Note:** this step is not necessary with Universal Edition, since with Universal Edition the same serial number that licenses Manifold System will also automatically turn on Business Tools, Geocoding Tools and Surface Tools.

- **When you are ready to permanently install Manifold on a particular computer system,** you may activate Manifold via the Internet automatically or you may activate manually. Note that although you **could** use the system to activate one Manifold license on several machines, it is against your license to attempt to use Manifold on more than one machine at the same time. Doing so could result in your serial number being permanently revoked. It is also risky to use up activations intended for re-installation in case of hard disk crash or machine failure, because if you use up your activations you might not have a spare activation in case of machine failure when you need it.

- **If you have licensed any extensions, such as Business Tools, Geocoding Tools or Surface Tools,** when you are ready to permanently install the extension on a particular computer system, you may activate the extension via the Internet automatically or you may activate manually. **Note:** this step is not necessary with Universal Edition, since with Universal Edition by activating your Manifold System license you will also automatically activate extensions.
Please take care to keep track of your serial numbers. For example, do not confuse a Business Tools, Geocoding Tools or Surface Tools serial number with a serial number for your main Manifold license. An Activation key fetched for a particular serial number will work with that serial number only. Do not try to use an Activation key that was fetched using one serial number with a different serial number. Such errors are easy to avoid if you keep track of your serial numbers.

**When No Direct Internet Connection is Available**

If your machine is not connected to the Internet, you will want some other way of visiting the Activation web site to get an Activation key with your serial number and System ID.

If you do not have a direct Internet connection for the machine on which Manifold is installed, you can use some other machine to visit the Activation web site. Although it may be tempting to simply print out the serial number email, use a different machine to visit the Activation web site and then manually write down the resulting Activation key, it is usually a bad idea not to use Copy and Paste because of the risk of typographic errors introduced through a lot of manual handwriting. Serial numbers, System IDs and Activation keys are so long that manual entry is begging for typographic errors.

Instead, take a moment to copy your serial number and System ID into some electronic form: write the serial number and System ID into a text file that you put on a USB drive or other removable medium, or email them to yourself using some free web email service such as hotmail or gmail that can be accessed from other machines. You can then go to a machine with Internet access and open that removable medium or email, copy and paste the serial number and System ID into the Activation web site to get an Activation key and then copy and paste the Activation key back into the removable medium or reply email. Back on the installation machine you can then easily copy and paste both serial number and Activation key from the removable medium or the reply email into Manifold's Activation dialog.

If you will be traveling to locations with no Internet access whatsoever, please plan ahead and activate your copy of Manifold before departing civilization. In extreme cases, people have used satellite phones or radio to contact colleagues back in civilization, they have read to them their serial number and System ID for use by the colleague, and then finally they have gotten their colleague to read back to them an Activation key. This is obviously a formula for errors, so plan ahead to avoid such situations.

**The Activation Dialog**

The first time that Manifold launches after installation, the **Activation** dialog will open to allow you to enter the serial number for your license and, optionally, an Activation key. If you enter just a serial number Manifold will run just fine so long as it is within 30 days of the date the serial number was issued. If you enter both a serial number and an Activation key, Manifold will be permanently installed.

When the **Activation** dialog opens, it automatically shows the **System ID** computed for your machine by Manifold. The **System ID** is a unique "fingerprint" for your computer that identifies a particular activation request. Making major changes to the computer, such as replacing a hard disk, installing a new version of Windows or changing a motherboard may change the **System ID**, or they may not. Such changes may occur over the life of a computer or Manifold license (neither of which will run to the end of time), so you can obtain five activations per serial number to enable you to re-install Manifold after making such major changes that may be required by hardware failure or by upgrades to new computer equipment.

If you enter only a serial number and it is not yet 30 days past the date of issue of the serial number, Manifold will continue to function without activation until 30 days later. There is no limitation of functionality, no limitation on the number of launches or anything like that: it is the complete, full product operating with all features operating for the edition that has been licensed. Each time Manifold launches, the system will pop open the **Activation** dialog to let you know how many days remain in the preliminary installation period and to give you the opportunity to activate Manifold should you choose to do so.

When the initial 30 day period expires, Manifold will not launch unless it has been activated. Because activations are a limited resource, the purpose of the preliminary 30 day installation period is to allow users to try out different hardware configurations before they decide which computer system will be their first choice for the Manifold license.

Take your time and resolve all hardware issues during the preliminary installation period. When ready, you can activate Manifold System on the computer you intend to use for Manifold.

When entering a serial number or activating via the **Activation** dialog you must have Administrator privileges. When an Administrator enters a serial number or Activation key into the **Activation** dialog, Manifold will be
activated for all users on that machine. To make sure you have Administrator privileges login as "Administrator." Don’t assume your login has full Administrator privileges.

When running the Activation dialog, make sure no other programs are running in background except standard Microsoft Windows processes and services. Do not run anti-virus checkers, uninstallers, registry "protectors," network checkers, special administrative protection programs or other such programs. These may interfere with the activation and permanent installation process.

If desired, the Activation dialog also may be launched via the Help - Activate command.

**About Serial Numbers**

Every Manifold System license is identified by a unique, permanent serial number. The serial number encodes the Manifold System edition that is licensed and the date and time the serial number was issued. It does not encode the user, it is not tied to a particular machine and it is not tied to a particular download, so be careful not to give it to people you won’t want to be able to use it.

Serial numbers are long alphanumeric strings that look like:

```
384C86E77919-4DAEA5A0-2BC41719A85FC5F1EA49A6243055
```

For a serial number to be valid, it must be reproduced **exactly** as it was issued by manifold.net. Do not change upper case to lower case, remove hyphens, add space characters or make any other changes. A serial number must be entered as a single line without any paragraph breaks. If your email system splits long lines into two lines, make sure to reassemble the serial number as a single line when entering it into the activation web site or into the Activation dialog.

**Note:** Manifold serial numbers do not end in a series of "X" characters. If you are trying to use a serial number that ends in a series of "X" characters, you are wrongly attempting to use a masked version of the serial number that has had the last few characters concealed with "X" characters to prevent theft. Use the original serial number sent to you, not the masked version.

When a Manifold license is acquired from manifold.net, a new unique serial number is generated. The serial number will be sent to the licensee by email using the email address provided at the time of purchase. If you acquired your license directly from manifold.net, you have been emailed the serial number for your license. If you have not received the serial number by email, please see the Tech Tip below to determine what problems in your email system may be preventing you from receiving your serial number email message.

Serial numbers have four functions:

- To identify the Manifold license acquired.
- To enable up to five activations of the Manifold license, accomplished by obtaining up to five Activation keys from the Manifold Activation key server.

Your serial number specifies which edition of Manifold System has been licensed. Professional Edition licensees will have been issued a serial number that specifies Professional Edition, while Enterprise Edition licensees will have been issued a serial number that specifies Enterprise Edition.

All Manifold editions are installed from a single installation file, either for 32-bit or for x64 Manifold systems. The installation file includes all the code necessary to run any Manifold edition as well as any optional extensions available for that product. All licensees use this same installation file. After installing Manifold System the type of serial number provided to the Activation dialog will control which Manifold System edition will be launched. In addition, providing a serial number for an optional extension in the Help - Activate Extension dialog will turn on that optional extension.

Providing a Professional Edition serial number in the Activation dialog will cause Manifold to launch in Professional Edition mode. Providing an Enterprise Edition serial number will cause Manifold to launch with all Enterprise Edition features enabled. Providing a Universal Edition serial number will launch Manifold with all
Enterprise Edition features and also with Business Tools, Geocoding Tools and Surface Tools extensions enabled as well.

Serial numbers may be revoked by manifold.net if the terms of the EULA are violated. In particular, disclosing your serial number to any third party (except for the sole purpose of a one-time transfer of your license to someone else) will result in the cancellation of a serial number. If your serial number appears in any hacker lists or hacker web sites for unauthorized use of Manifold System it will be cancelled immediately. Therefore, please take care to keep your serial number secret.

Serial numbers are very important so do not lose your serial number. You cannot install Manifold without a serial number. When you receive your serial number by email, print a copy of the email and save the printed copy in a safe place. It might not be possible for manifold.net to recover your serial number if you lose it. If it is possible to recover your serial number, manifold.net will charge you a fee to recover it.

30 Day Preliminary Installation

Each serial number encodes the date it was issued as well as the period of time after that date during which Manifold System will function without activation. The limited, 30-day period of time during which Manifold will function with a serial number alone is called the preliminary installation period. The preliminary installation period runs 30 days from the date the serial number was issued. There are no limitations on the functionality of the software during that period. It is not some limited "trial edition" period or "demo" period - it is the fully complete, full power, fully operational software during that period, right from the very first day with no activation or other special procedure required.

For example, if you acquired your Manifold license from manifold.net on July 8, a serial number will have been generated on that date and sent to you. When entered into the Activation dialog, that serial number will allow Manifold to launch without activation until August 8. After August 8, the preliminary installation period will have ended and activation will be required to allow Manifold to launch.

Because serial numbers are very long, to avoid typographical errors use Windows Copy and Paste to transfer them to the Activation dialog as follows: Open the original email in which you received notice of your serial number. Highlight the serial number and choose Edit - Copy (or press CTRL-C, the Windows keyboard shortcut for "copy"). Launch Manifold and right-click into the Serial number box in the Activation dialog and choose Paste from the context menu (or, click into the box and press CTRL-V, the standard Windows keyboard shortcut for "paste"). This procedure will copy the serial number from the email and paste it into the Activation dialog without the need to manually type a long serial number.

Just like a Manifold System license, extensions such as Business Tools, Geocoding Tools and Surface Tools also use serial numbers that offer a period of preliminary installation after which activation is required for permanent installation.

Display of Serial Numbers in the Activation Dialog

During the preliminary installation period whenever Manifold is launched it will display the Activation dialog to remind you that Manifold has not yet been activated.

The Activation dialog will display the serial number used for preliminary installation, but the last few characters will be replaced with "X" characters to keep your full serial number secret. This is similar to how credit card numbers are often partially masked with * or other characters on receipts so that the receipt cannot be used to steal the credit card number, yet the credit card owner can see enough digits of the actual number to determine which credit card was used for the transaction.

If your serial number was not masked with X characters, anyone who had physical access to your machine could launch Manifold, copy the serial number and then steal it. The version masked with X characters is useless for activation, but it still contains enough unique digits to determine which serial number is in use. Manifold licensees who have many serial numbers can thus easily determine which serial number was used on which machine.

Do not let this simple privacy measure cause confusion when later activating your license. Just as the masked version of a credit card number printed on a receipt cannot be expected to work if you copy the * asterisk characters and attempt to use that number to buy something, just so the masked version of the serial number will not be able to fetch an Activation key.
When copying and pasting the serial number during activation, make sure to use the **full serial number** as originally emailed to you. Do **not** use the masked version of the serial number that appears in the Activation dialog.

**About System IDs**

The **Activation** dialog will always show the same System ID for your computer unless a major change is made to the computer's hardware or software. The System ID is a long alphanumeric string that looks like:

906578E309D5FFDCB994F8D2A8C47BCE113AE103D

If we activate Manifold via the web we don't have to worry about handling System IDs. If we choose to manually activate Manifold, we will need to copy the System ID from the Activation dialog to use at the Manifold web site for activation. The **Activation** dialog will let us use **Copy** to copy the System ID from the box in the dialog.

**Note:** The System ID box is read-only in the **Activation** dialog because the System ID is always generated by Manifold to uniquely identify an activation request.

**About Activation Keys**

An Activation key is required to permanently install Manifold to your computer system. Each Activation key is unique and matches only the one serial number and the one System ID used to obtain it.

Activation keys are very long because they encode three important pieces of information:

- The date and time the Activation key was issued.
- The serial number used to obtain the Activation key.
- The System ID used to obtain the Activation key.

Activation keys are very long alphanumeric strings that look like:

B8384D8A1C6B-4942B549-2911DE16722F3727080800EE2D2B74EC02CC5F1EA6025FECEAE

[Activation keys do not include returns - An Activation key is entirely the same line. The Activation key shown above is too long to fit into the margins of this page without a line wrap.]

Any user with a valid Manifold System serial number or with a serial number for an extension such as Business Tools, Geocoding Tools or Surface Tools may obtain an Activation key from manifold.net subject to the following conditions:

- When an Activation key is issued, it remains valid for only 72 hours after it is issued. If it is not used in the **Activation** dialog to permanently install Manifold within that 72 hour time period, it becomes useless and will no longer function to install Manifold System.
- An Activation key is valid only for the serial number used to obtain it. For example, an Activation key obtained with a Surface Tools serial number cannot be used with a Manifold System serial number. An Activation key obtained with one Manifold System serial number cannot be used with a different Manifold System serial number.
- An Activation key is valid only for the System ID used to obtain it. For example, if we obtain an Activation key generated by one System ID we will not be able to use that Activation key to activate Manifold using a different System ID.
- Each serial number can have a maximum of **five** Activation keys issued for it. The Manifold key server keeps track of how many Activation keys have been issued for each serial number. After five Activation keys are issued for a serial number, no more Activation keys will be issued. Five Activation keys are allotted per serial number in case you need to re-install Manifold at a future date as a result of system upgrades or other changes. Five Activation keys are **not** issued to enable users to violate the license by installing the same Manifold license on five different machines.
- You must have logged in as **Administrator** in Windows Vista and your login must have Administrator privileges in Windows 7, Windows Vista, Windows Server 2008, Windows Server 2003, Windows XP or Windows 2000 when launching Manifold to provide an Activation key via the Activation dialog.
Obtaining an Activation Key

The fastest, easiest way to obtain an Activation key is to launch Manifold and in the Activation dialog provide a valid Manifold serial number and then press the Get Activation Key via the Web button. Manifold will automatically reach out through Internet and get an Activation key for us. It is madness to use manual activation if our computer is connected to Internet.

If desired, we can always obtain an Activation key manually. To do so, visit the Manifold Activation key web site at

http://www.manifold.net/activation

This website requires a modern browser that can correctly handle JavaScript. You must have scripting enabled in your browser. If you have any difficulties using this website, please visit it using Microsoft Internet Explorer 5.5 or more recent edition of IE. Since you must have IE installed on your computer (even if you do not use it as your default browser) to run Manifold, you will have an IE installation available to use even if you normally use some other browser.

Visit the Manifold Activation key web page and provide a valid Manifold System serial number and the System ID for the system on which the Activation key will be used. If the serial number has not yet been used to generate the five Activation keys it is allowed, the key server will provide an Activation key. Immediately write down the Activation key in a safe place. Because the key is very long, the safest way of writing it down is to use Windows Copy and Paste to save it in a text file.

To use Copy and Paste to save the key in a text file, after the key server provides an Activation key highlight the Activation key on the web page with your mouse and choose Edit - Copy in Internet Explorer (or press CTRL-C on the keyboard) to Copy the key. Next, open a text file on your computer using Notepad or Microsoft Word or any other convenient text editor and choose Edit - Paste (or press CTRL-V on the keyboard) to Paste the key into the text document. Save the document. Save a copy of the document in a safe place, such as on a different disk drive or print out a hard copy on paper.

Once you have obtained an Activation key you can launch Manifold and provide the Activation key to the Activation dialog to permanently install Manifold System. Since the Activation key is very long, the best way to avoid typographic errors is to use Windows Copy and Paste to enter the Activation key into the Activation dialog. Open the document in which you saved the Activation key. Highlight the key and Copy it. Click on the Activation dialog and right-click into the Activation key box and Paste the key.

Because Activation keys have a limited life of only 72 hours, do not obtain an Activation key until you are ready to use it to permanently install Manifold System. Once you have obtained an Activation key, you must use it within 72 hours.

Important: Using the same serial number within any 24 hour span to fetch an Activation key, even if the System ID is different, will result in the same Activation key. This is both a safety measure and an anti-piracy measure. It is a safety measure so that people who mindlessly push buttons multiple times on web sites or dialogs don’t use up all their Activation keys in a matter of seconds, and it is an anti-piracy measure to discourage cryptographic attacks.

Important: The above means that you should wait at least 24 hours after getting an Activation key for your desktop machine before getting an Activation key using the same serial number to activate a Manifold license installed on your portable computer. If you don’t wait 24 hours, you’ll get the same Activation key back even though you used a different System ID for the portable computer. The resultant Activation key, therefore, won’t work. This problem is easy to avoid: simply wait at least 24 hours after using a serial number to activate Manifold on your desktop machine before using the same serial number to activate Manifold on your portable computer.

Administrator Privileges Required

As mentioned earlier, you must have Administrator privileges to use the Activation dialog. “Administrator” privileges are special rights granted to the Administrator user and to members of the Administrators group in Windows operating systems such as Windows 7, Windows Vista, Windows Server 2008, Windows Server 2003, Windows XP or Windows 2000 that provide user logins. If using XP Home in default configuration any user login with “Computer administrator” rights is OK.

If using Windows Vista, you must login using the actual Administrator account. Do not use some account thought to have Administrator privileges. Use the actual Administrator account. Microsoft will hide this
account in some versions of Vista, such as Vista Home, they apparently think might be used by people too stupid to be trusted to have full control over their computers. Can’t blame them. See the Vista Admin Login page in the Support page on the manifold.net web site.

Administrator privileges allow users to change anything in the system. Ordinary users do not normally have the required rights to make important changes to the system such as may be required when installing new software.

If you work in an organization where someone else configures and administers your computer, the login name you use might not have Administrator privileges on your computer. In such cases you should ask someone with Administrator privileges to login to your computer and to use their Administrator account to install Manifold System, including either preliminary installation with a serial number or permanent installation via activation.

Caution: If you do not have Administrator privileges the Activation dialog will accept the Activation key and will allow you to run for that session, but the installation will not be permanently accomplished. This feature enables emergency activation of Manifold on a temporary basis if an Administrator is not available. If you do not remember to use the Activation key with an Administrator login during the 72 hour life of the Activation key you will have wasted that Activation key. If you do not have Administrator privileges when the Activation dialog is launched, Manifold will warn you that any activation will be good only for that session. Don't make the mistake of ignoring such warnings!


Do Not Waste Activations

Five activations may seem like a lot, but they are not so many that you can afford to waste activations, even if you expect to get a fresh batch of activations every year or two through low cost upgrades, by free activations handed out from time to time or as a result of purchasing additional activations.

Over the course of two or three years it is quite possible that you might want to upgrade your computer's hard disk, install a new version of Windows, or otherwise make changes that require re-installation of Manifold System. Each installation requires activation, using up one of the five activations allowed. Because each Activation key works for only 72 hours and works only for the System ID used to fetch it, when you need to reinstall Manifold over the course of months or years you will have to use up one of the allowed activations for each installation.

If you upgrade the hard disk once and upgrade Windows once, you will use a total of three activations (one for the original installation and one each for the disk and Windows re-installations). That will leave you with only two activations available should you ever need to re-install Manifold due to unforeseen circumstances, such as a disk crash, lightning strike, power spike or other calamity.

Five activations are plenty for even very active computer users but they are not enough to waste activations thoughtlessly or to give out free copies of Manifold to friends. If you are not careful with activations, you might not have any left when you really need one.

Note that since every serial number can be used to obtain only five activations it is important to keep your serial number a secret. If anyone else learns your serial number, they could use your serial number for their own use in activation Manifold. If your serial number ever gets out to a hacker your five allowed activations will be gone overnight.

As noted earlier in this topic, if you take advantage of upgrades to new releases, you can get a new serial number and five fresh activations for a low cost. A single Manifold serial number can allow you to run your license for years, but given advances in computer hardware and software (as well as in Manifold) most people will take advantage of upgrades to renew their serial numbers from time to time.

Once installed a Manifold license will run as long as that computer continues to operate in sufficiently unmodified form to not require re-activation. The five activations provided for a serial number enable re-installation and re-activation within a reasonable period of time; however, re-activation of old or obsolete licenses may not continue to be supported once future versions of Manifold are released. Although manifold.net may voluntarily support use of activation keys for outdated versions of Manifold for some reasonable period of time (such as three years) licensees should not expect such support.

System Date Must be Accurate
Serial numbers and activation keys encode the date they were issued. They will not work with system dates that are obviously wrong such as a system date before the date of issue of the serial number or Activation key. Permanent installation also encodes the date as well. Your system date and time must be reasonably accurate (within the correct day of the date) for activation to work.

If after activation Manifold suddenly raises the **Activation** dialog again and no major hardware or software changes have been made, check the system date. If you or someone else has reset the system date to an obviously wrong date, Manifold will not launch.

**Manifold Removal and Re-Installation**

Once your Manifold System license has been activated on a particular computer, you may uninstall Manifold System using the Windows Control Panel - Add/Remove Programs applet and then later re-install your Manifold System license on that same machine without requiring any new activation. The computer system will "remember" that your Manifold System license has been activated. For example, if you installed Manifold System from the original release first issued you could uninstall that original release and then download and install an update for that same version without needing to activate again.

This doesn't work, of course, if you make changes that would destroy the activation status on your machine in any event. For example, if you uninstall Manifold and then change your hard disk or install a new version of Windows when you re-install Manifold version you may need to use an Activation key.

Note that this applies only to that same Manifold version and updates or service packs published for that same version. For example, if you have a Manifold 8 license you cannot download and install an update published for some future version such as a future Manifold product and expect that the new product will be functional. If you want to move up to a newer Manifold version you'll have to acquire a license for that newer version.

**Activating an Extension**

See the Installing and Activating a Manifold Extension topic for details on activating a Manifold extension such as a Business Tools, Geocoding Tools or Surface Tools extension.

**Installation on a Notebook Computer**

The Manifold End User License Agreement allows you to install your Manifold license both on a desktop machine and also on a portable computer used by you, so long as the two computers are not used at the same time.

Activating your license on a desktop computer and also on a portable computer will **use up two activations**. Two activations are required because each individual activation, that is, each individual Activation key, will get fetched by a different System ID that identifies that particular activation request.

It is true that installing Manifold both on a desktop computer and also on a notebook computer will consume activations at a higher rate than installation only on one computer; however, that's the choice of the user. If you choose to install both on a desktop computer and also on a portable computer, make sure to take advantage of any special upgrade offers to avoid running out of activations.

**Important:** You should **wait at least 24 hours** after getting an Activation key for your desktop machine before getting an Activation key using the same serial number to activate a Manifold license installed on your portable computer. If you don't wait 24 hours, you'll get the same Activation key back even though you used a different System ID for the portable computer. The resultant Activation key, therefore, won't work. This problem is easy to avoid: simply wait at least 24 hours after using a serial number to activate Manifold on your desktop machine before using the same serial number to activate Manifold on your portable computer.

**Example 1**

John Doe orders Manifold System via the Manifold Online Store on July 7. He receives an email with his serial number when his order is processed on July 8. A few days later he downloads the Manifold installation. He is then called away on a business trip before he has a chance to install Manifold. He is not able to install Manifold System until 22 July. John logs in as **Administrator** for all of his installation and configuration activity with Manifold.
When he launches Manifold for the first time on 22 July he is not sure exactly what to do about activation so he copies the serial number from the email sent to him and uses it to launch Manifold without activating his license. He feels he has plenty of time until August 8 to figure out what he wants to do before he needs to activate his license.

John works with several different computers. He installs Manifold on several different machines during his first few days of use to see if he needs to make any performance upgrades. He installs Manifold on his notebook computer as well. All of his machines run Windows Vista except his notebook computer, which still runs Windows XP.

After a few days he decides to upgrade his notebook (finally) to Windows Vista. To assure a clean install of Windows he reformats the hard disk and installs Windows Vista and then he begins reloading his software. He re-installs Manifold, using his serial number. Because it is not yet August 8, John can continue using his serial number to make as many preliminary installations as he wants without needing to activate his license.

As August 8 draws near, John decides that he will use Manifold on his notebook computer and on his new desktop machine at the office. Every day that he launches Manifold the Activation dialog reminds him that he has only a few days left before he must activate his license. However, John is a procrastinator and gets involved in another project for a few days.

The next time he works with Manifold is on August 12, a few days after the preliminary installation has expired. When Manifold launches it tells John the preliminary installation period has expired and asks him to provide a serial number and Activation key. John realizes it is time to stop procrastinating and to activate his license.

John launches Manifold on his desktop machine and activates his license via the web. He would like to use Manifold right away on his notebook computer, but he remembers reading the portions of this help topic that start with "Important" in bright red letters. He waits a day and a half to be sure he is past the 24 delay period required to get a new Activation key using the same serial number.

He then launches Manifold on his notebook computer and realizes that his notebook does not have a web connection. If his notebook computer had an Internet connection he could have activated his license over the web. Instead, John will have to do a manual activation.

John copies the System ID from the Activation dialog and pastes it into a text file that he saves to a zip drive plugged into one of his notebook computer's USB ports. John unplugs his zip drive from the notebook computer and plugs it into his desktop computer. He opens the text file and copies the System ID. John visits the Manifold activation web site and provides his serial number and the System ID for his notebook computer. The activation web site obligingly spits out an Activation key. John copies that Activation key to the text file in the zip drive along with a copy his serial number.

John knows the Activation key is good for only 72 hours so he does not procrastinate this time. He makes use of it right away. John plugs the zip drive back into the notebook computer and copies his serial number (copied from the original email) together with the new Activation key into the Activation dialog to permanently install Manifold on the notebook.

He now has a permanent installation on the two machines (a primary desktop machine and an optional portable computer) allowed him by the Manifold License. After all this he has used two activations. He still has three activations left on his serial number should he ever need to reinstall Manifold.

Because John doesn't want to get confused about where his licensed software is residing (and because he's straight about honoring his software licenses) he takes a moment to uninstall Manifold from all the different systems he installed it to while experimenting during the preliminary installation period. No serial number or key is required to install or to uninstall Manifold System. A serial number or activation is required only to launch Manifold.

After a year, John decides to upgrade to Windows 7. John installs Windows 7 on his desktop computer and then (logging in as Administrator) re-installs Manifold System as well. He activates that Manifold installation. He still has two more Activation keys in case he ever needs to install Manifold again. For now, he does not install Windows 7 on his notebook computer.

After another year, John purchases a new notebook computer with a hot new multi-core processor, built-in broadband cellular Internet and Windows 7 pre-installed. He installs Manifold on the new computer. He now has only one original activation left on his serial number, but he knows he gets a free activation ever year and he can always buy more activations at a low price. John's not worried because he knows he will soon upgrade to the next edition of Manifold System, which will get him a new serial number and five more activations.
After a few more months John decides to acquire a Surface Tools license. He orders Surface Tools via the Online Store and the next day receives an email giving him a serial number for Surface Tools. Because John already has decided the computer on which he will run Surface Tools he decides to activate the package right away. He logs in as Administrator, launches Manifold, launches the Help - Activate Extension dialog and copies and pastes the Surface Tools serial number into the dialog and then activates over the web. After restarting Manifold he is pleased to see the new Surface Tools features have been enabled.

Example 2

Henry Doe, John's younger brother, a student, also acquires a copy of Manifold via the Online Store. He pays no attention to the email he receives except to note the serial number. When Manifold arrives he installs it and then copies and pastes the serial number. He does not activate the license.

After a few weeks he decides it is time to get activated, but he does not "trust" web-based activation and decides to manually get an Activation key from the Manifold activation web site. When the Activation dialog pops up he copies the serial number from the dialog and attempts to get an Activation key on the web site. This does not work because Henry has copied the masked version of the serial number from the dialog, which has X characters substituted at the end. He is not using the original serial number sent to him by email. Henry, of course, has not read this topic so he thinks it is just fine to copy the masked version of the serial number from the Activation dialog.

Henry emails tech support and swears he is copying and pasting the serial number accurately. Tech support responds by asking Henry a series of detailed questions, some of which check up on typical errors (one question asks if Henry copied the serial number from the original email or from the Activation dialog). Henry is offended that they would ask him to confirm such simple things so instead of reading the questions and responding accurately, Henry just skims the questions and emails a testy reply, "Yes, I'm doing all that."

Manifold's experienced support personnel can tell right away that Henry isn't very good with details, so they ask him each question individually again in a different way and they ask him to give the exact serial number he is using. As soon as they see the X characters at the end of the serial number they solve the problem. Henry copies the serial number from the original email, gets an Activation key and activates Manifold. This experience transforms Henry for the better.

Henry realizes that perhaps things will go better if he reads the manual instead of trying to "wing it" all the time. He reads Manifold documentation every night and becomes an expert. His grades soar from C's to A's. He uses his newfound Manifold expertise to invent a new method for finding diamond-bearing Kimberlite pipes. He stakes claims on fabulously rich diamond properties and becomes very wealthy. He marries a beautiful Hollywood actress who is entranced by his continued hobby of reading Manifold documentation in depth. His friends admire his ability to cite any Manifold Help topic from memory. [This paragraph contributed by Manifold tech support].

Example 3

Jane Doe orders Manifold System via the Online Store. She receives an email providing her with a serial number. Jane downloads the Manifold installation from the Product Downloads page and installs Manifold. The first time she launches the product, the Activation dialog pops up asking her for a serial number and Activation key. She has forgotten about the serial number email message, but quickly finds it in her email inbox and pastes it into the Activation dialog. After a few more days she decides to permanently install the package so she activates Manifold over the web.

After a few weeks of work, Jane encounters problems with her Windows system when an application she recently installed is not working properly. She decides to simplify her system to see what could be causing the problem by uninstalling some of the applications she has installed recently. She uninstalls Manifold using the Windows Control Panel Add / Remove Programs dialog. Removing Manifold proves it is not the problem. After removing a few other applications she traces the problem to a freeware screen saver utility. After cursing lame freeware authors she removes the offending utility.

To re-install Manifold she installs it again from the installation file she previously downloaded. When she launches Manifold, it automatically launches without the need to enter a serial number or an Activation key since Manifold has already been permanently installed on this computer system. If Jane had re-installed Windows or made other substantial changes (such as rolling back her registry or other major alterations), she might have had to use an activation to re-activate Manifold System, but since she has not made any major changes everything works automatically for her.
A month or two later when Manifold launches it tells Jane that a new update is available. She downloads the update, uninstalls the previous version of Manifold and then installs the update following the update's installation instructions. Manifold launches without opening any activation dialog. Once again there is no need for Jane to enter a new serial number or to re-activate Manifold.

Internet Map Server

When Manifold is running as an Internet Map Server (IMS) it will not raise the Activation dialog. Therefore, before attempting to use Manifold in IMS mode you must launch Manifold interactively at least once to provide a serial number for preliminary installation, or a serial number with activation for permanent installation. Alternately, you may use command line activation (see below) to provide a serial number and Activation key.

Runtime Licenses

Every Manifold installation will launch with the Activation dialog when it is first launched. When a serial number is provided during that first launch the Manifold installation will become the edition designated by that serial number. Providing a runtime serial number will configure that Manifold installation as a runtime edition.

Because runtime licenses do not have a GUI once they are configured with a serial number they are normally activated when Manifold is first launched. After a runtime license is configured with a runtime serial number, Manifold will not launch interactively so the Activation dialog will not appear during the 30 initial days of operation on serial number alone.

After the first 30 days if a runtime license running on serial number alone is launched then the Activation dialog will be raised. This will enable re-entry of the serial number and activation. However, because it is a messy situation to launch a runtime license with a serial number alone and then have to wait for up to 30 days to get the Activation dialog back again, to avoid such hassles people will normally activate a runtime license the first time Manifold is launched. Strictly speaking they don't have to, but it is such a common sense convenience that's what most people do.

Command Line Activation

Manifold may be activated from a Windows command line. Command line activation options are a relatively limited tool and are only intended to be used to switch editions of Manifold that are installed on the same machine or to force a non-interactive edition (such as a runtime or a License Server installation) to raise the Activation dialog.

Command line activation allows some administrative tasks to be accomplished more easily:

- IT administrators may switch editions of Manifold installed on machines in their network using automated tools, without requiring manual action by users. However, most IT administrators will use the Manifold License Server product to simplify life even further.
- Application developers or Manifold IMS administrators using runtime editions or IT administrators using License Server editions may switch the edition of Manifold installed on a machine from a non-interactive license to a regular, interactive license, such as Professional Edition or Enterprise Edition.

There are two command line switches for activation. One switch is for activation of Manifold System itself and the other switch is used to activate an extension:

```
manifold.exe /activate:serial,key
```

Activate Manifold System.

```
manifold.exe /activateExt:serial,key
```

Activate an extension.

In the above, `serial` is the serial number and `key` is the Activation key. Because a System ID must be noted in order to get an Activation key with a serial number, the user must be able to launch Manifold interactively at least once on a machine to obtain the System ID.

Supplying an invalid serial number or a valid serial number without an Activation key will pop open the Activation dialog. This is a handy way of forcing a non-interactive Manifold installation to raise the Activation dialog so that the System ID may be obtained. For example, using the command line switch:
...will pop open the Activation dialog since "dialog" is not a valid serial number.

While command line activation options can be used to automate the deployment of Manifold licenses within an organization, Manifold System License Server provides a significantly easier and much more efficient way of doing so. The License Server allows installation of Manifold without individual activation. Instead, licenses are authenticated by connecting to the centralized License Server using the pool of available licenses.

See the Command Line Options topic for other command line options.

Frequently Asked Questions (FAQ)

Someone else got my Manifold license for me and manifold.net tech support will not assist me with activation. How do I get assistance with activation? Contact whoever got your license for you for assistance. manifold.net will only support the original licensee using the original email to which serial numbers were sent. There are two main reasons for this. First, serial numbers and activation keys are confidential information and manifold.net will not risk assisting anyone who may be using someone else's license without authorization. Second, manifold.net provided important information about activation to the original licensee but has no control over whether the original licensee passed on that information to you. If you are in an organization the designated licensee is the sole and exclusive contact with manifold.net.

How do I transfer my Manifold license from one computer to a different computer? Login as Administrator for all of the following steps: Step 1 - Make sure you have your original serial number email so you can copy and paste your serial number without typographical errors. Step 2 - Check the serial number on the serial number status page in the Manifold support web page to verify you have an activation remaining. Step 3 - Uninstall Manifold from the old computer using the Windows Control Panel's applet for uninstalling programs. Step 4 - Download the latest Manifold installation file for your version of Manifold onto the new computer and install it. Step 5 - Launch Manifold and provide your serial number, getting an Activation key.

Is my Manifold license locked to me? Can I transfer my license to another user? Manifold installation files downloaded from the website, licenses, serial numbers or activation keys aren't locked to or bound to or identified with any specific user. That makes it easy for companies and resellers to procure software for distribution within their companies or to their clients. But that also means that anyone who has a serial number for a license can use it, so be careful not to give out a serial number to anyone who you don't trust to use it. But that also makes it easy to transfer a license you have obtained: simply uninstall Manifold from your computer and then give your serial number to the person you want to have it. The transfer is then just like transferring a Manifold license from one computer to a different computer. Do not continue using Manifold because if the same serial number is run simultaneously on multiple computers that is a violation of your license and could result in the serial number being revoked automatically.

Must I activate Manifold to get a fully operational license or to remove limitations in the license? No. There is no such thing as a limited Manifold license that is a "demo" or "trial" license with any sort of limitation. Every license issued by Manifold is a full-power, fully complete license right from the very first day and requires no activation for fully functional operation. If you have any difficulty operating the software, don't think you need to activate it to turn on some feature that appears not to be running. That's not the problem (to find the problem, consult the user manual or contact tech support for assistance). Activation is purely a mechanism that allows easy experimentation with different computer configurations during the first 30 days of a license's life while discouraging piracy thereafter, and it provides a way of keeping track of which licenses, for those which are refundable under the limited 30-day money-back guarantee, might still qualify for a refund. For the first thirty days of every license's life no activation is required. You may install it on however machines you like during the first 30 days of the life of a license. There is no need to activate to "turn on" any feature or to remove any limitation. Every license is full power, fully functional, all the time. Note that this is a very different approach than some software, which provides a limited "trial" software version that has some restricted or missing capabilities, such as disabled printing or disabled file saving or some limited number of launches or some other limitation, until an activation code or other authorization code is provided. Manifold software is always fully functional all of the time.

Can't I just keep setting my system clock back every 30 days to run forever without activation? Sure, if you don't mind doing that. Manifold will run forever in that case as a fully operational license with no limitations and no need to activate. Some users do just that if they find Activation is inconvenient, or just if they want to try out different combinations of hardware. Manifold uses a relaxed approach to copy protection that is not as harsh as some schemes that won't allow such relaxed, full-power operation without utilizing a specialized licensing procedure or which limit the software to a limited number of launches, a limited number of saves or otherwise limit functionality. However, you have several other options. You can simply reset your system clock that is something easy to get confused about or it may interfere with the operation of other software that takes a harsher approach to copy protection. It's not a good idea to use this method to share one Manifold license with your friends or to try to run one Manifold license on many machines because if the same serial number is run simultaneously on multiple computers that is a violation of your license and could result in the serial number being revoked automatically.
Activation and Licensing

I have licensed a 64-bit Manifold edition. Can I use my 64-bit license serial number to activate a 32-bit Manifold installation? Yes. The serial number you get for a 64-bit Manifold license can activate either a 64-bit or a 32-bit Manifold installation. For example, suppose you have 32-bit Windows computer today but you know in a few months you'll be upgrading to 64-bit Windows. You could use a 32-bit Manifold installation file to install 32-bit Manifold on your 32-bit Windows system and use the serial number from a 64-bit Manifold license to activate it. In a few months when you upgrade your computer to 64-bit Windows you could use a 64-bit Manifold installation file to install 64-bit Manifold on your 64-bit Windows system and use your 64-bit Manifold license serial number to activate it.

I acquired my license and received a serial number more than thirty days ago. Can I still install Manifold System? Yes. Serial numbers don't expire over time. You can run Manifold System but you will need to activate your installation since the time period available for preliminary installation has already expired. Launch Manifold and in the Activation dialog provide the serial number and then get an Activation key via the web. Done!

I received my serial number a long time ago and never got around to installing Manifold. Can I still activate my license? Yes. For the purpose of activation, serial numbers never expire. As long as you haven't used up the activations allotted to your serial number, you can activate the license using your serial number. Note that re-activation of old or obsolete licenses may not continue to be supported once future versions of Manifold are released. Although manifold.net may voluntarily support use of activation keys for outdated versions of Manifold for some reasonable period of time (such as three years) licensees should not expect such support.

After a previously successful installation and activation I need to activate again, but the System ID box is empty. Something has changed in your system that has made it impossible for Manifold to find a .NET assembly that needs to be used. Uninstall Manifold and then reinstall it. If it doesn't fix the problem, try repairing the Microsoft .NET framework using the Microsoft control panel applet, and then uninstall Manifold and reinstall it.

I tried to get an Activation key on the Manifold activation web site with my serial number and the web site would not give me one. Check the obvious first:

- Did you correctly enter the serial number? Use Copy and Paste to avoid typographic errors. Include the hyphens, do not change upper case to lower case and make sure to otherwise enter the serial number exactly as it was given to you.
- Did you provide the System ID copied exactly from the Activation dialog for the target system?
- Copy the serial number from the original serial number email message. Don't copy it from the Activation dialog where it is masked off by XXX characters at the end so that it cannot be stolen.
- Did you obtain the serial number from Manifold? If you obtained your license and serial number from a third party it might not be a valid serial number.
- Has the serial number been used up? You can only get five Activation keys for each serial number at the activation web site. If you or anyone else has been getting Activation keys with that serial number, it could be used up. Keep in mind that if you or someone else has been fetching Activation keys using the Get Activation Key via the Web button, each time you activate Manifold in that manner you use up one of your five Activation keys.
- Has the serial number been revoked? If you did not pay your purchase order or have reversed ("charged back") your credit card payment to Manifold, your serial number has been revoked. As they say, “no pay, no play!” If your serial number has been published on Internet or has appeared in hacker lists of serial numbers used to steal software, it has also been revoked.
- Has your serial number been traded in on an upgrade? If you have upgraded from a 32-bit license to a 64-bit license the old 32-bit license is no more. An upgrade is not a free second license: it is a replacement for a previous license.
- Some serial numbers allow a limited period of operation without the ability to obtain an Activation key. Such serial numbers are issued to journalists or other reviewers to allow a limited period of operation. If you have such a serial number the key server will not give you an Activation key.
- Try using Internet Explorer 5.5 or 6.0 as your browser. Some other browsers have JavaScript errors without reporting those errors. Firefox may be a cool browser, but some versions of it are not able to deal with elementary javascript without errors.
- Make sure scripting is enabled in your browser. Some people turn off scripting for security reasons, but that prevents the Activation web site from working.
- Make sure you are using a PC. If you are using a Mac, make sure your browser settings are the same as would be used on Windows.

I tried to visit the Activation web site and got a Javascript error. Use a browser that functions correctly with Javascript. Since you must have IE6 installed for Manifold, we suggest using Internet Explorer. If you are using a non-Microsoft browser, try using IE.
If I accidentally press the "Create New Activation Key" button twice on the Activation Keys web page, will I waste an Activation key? No. Once an Activation key is generated for a serial number, if the same serial number is used again the web site will return that same Activation key throughout the first 24 hours of the 72 hour life of that Activation key. If you press the button several times you will get the same key back several times. This is a safety measure to avoid wasting keys. The same Activation key will be returned even if you use a different System ID with the same serial number so if you intend to install the same serial number on a desktop machine and also on a portable computer (the only licensed use of the same serial number on more than one machine), you should wait at least 24 hours between activating the desktop machine and getting an Activation key for the portable computer.

I entered my serial number into the Activation dialog but Manifold will not launch or reports that the serial number is invalid. Check the obvious first, including the Troubleshooting section at the top of this topic:

- Did you correctly enter the serial number? Use Copy and Paste to avoid typographic errors.
- Did you obtain the serial number from Manifold? If you obtained your license and serial number from a third party it might not be a valid serial number.
- Has the preliminary installation period expired? If it is after 30 days from the date the serial number was created and you don't want to reset your system clock accurately you will also need an Activation key to activate Manifold.
- Is your system date set accurately?
- Are you trying to use a serial number issued for an extension (such as Business Tools, Geocoding Tools or Surface Tools) to activate Manifold System itself, or vice versa are you trying to use a Manifold System serial number to activate an extension?
- If you are entering both a serial number and an Activation key, have you mixed up a serial number for one license with an Activation key that was fetched using a different serial number? Activation keys only work with the serial number with which they were fetched.
- A specific case of the above error: are you trying to use a serial number for a 32-bit Manifold product with a 64-bit Manifold installation? Running Manifold System in full 64-bits requires purchasing an x64 Manifold System product to get a 64-bit serial number. See the 32-bit and 64-bit Manifold Editions topic.
- Is the serial number for the same Manifold release that has been installed? You can't use a 7x serial number to launch an 8.00 installation, for example.

I entered a serial number and Activation key and Manifold worked OK, but now it does not. It is most likely that you did not use the Administrator login in Windows or that your login did not have Administrator privileges in Windows 7, Windows Vista, Windows Server 2008, Windows Server 2003, Windows XP or Windows 2000 when you used the Activation dialog to permanently install Manifold. The usual scenario is as follows:

- The user has a non-Administrator account that they use for daily operation.
- When first installing Manifold, the user reads all the material, logs in as Administrator and installs the product. When the Activation dialog launches, the user provides a serial number (only) and the product launches OK.
- Over the next few days and weeks the user logs in via the non-Administrator account and continues using the system in a routine way, including launching and using Manifold.
- One day when launching Manifold, the Activation dialog launches and announces to the user that activation is required. This happens because it has been 30 days since the serial number has been issued so now it is time to activate the license for continued operation.
- Right away, the user presses the Get Activation Key via the Web button to fetch an Activation key and presses the Accept button, ignoring all warnings about not having Administrator privileges.
- Manifold launches OK, but it is running in emergency mode for one session only.
- However, that previous step was an error because the user was logged in using his or her routine non-Administrator account. What has happened is that because it was several days or weeks after the initial installation, the user has forgotten the warnings to log in as Administrator when either a) installing Manifold or b) providing keys to the Activation dialog.
- Manifold will accept an Activation key in non-Administrator mode, but it only does so for that session as an emergency way of allowing Manifold to function even if the user cannot find someone right away with Administrator privileges.
- If you are sure it is not an Administrator privileges issue, something was changed in your system (Windows re-installation, restore from pre-installation image, hard disk changed, etc.). Someone might have tried to jump back to a previous Restore point in Windows, for example.

I activated Manifold on my laptop running XP and it ran OK for years but now it has suddenly stopped working and has raised the Activation dialog. I have installed no new programs or made any changes to my computer. What is the problem? Something has destroyed the activation status on your computer. The most common reasons:
• The system date has changed. Old laptops can have unstable clocks, so you might find your system date is not today's date. Reset the clock to today's date to solve the problem.
• A Windows restore has been done to a point before Manifold activation. Undo the restore.
• "I have installed no new programs or made any changes to my computer." Often said, but never believed by the veteran tech support engineer. People install new programs all the time. Every time you visit a web site with "active" content you install new programs, to name just one of millions of ways in which people install new programs without realizing it. Some such programs can have very destructive effects.
• There has indeed been a change in your Windows system that can affect activation status. For the same reason people don't realize when they get hit with a virus, it is easily possible that without realizing it you have installed some program or virus that has changed things. Some viruses make significant alterations in the registry and elsewhere as part of attempting to prevent anti-virus software from eradicating them, and such viruses can affect activation due to the havoc they wreak upon Windows.
• There has been a serious change in your hardware or intermittent failure of key components. Setting aside the obvious case of someone who changes hard disks, changes the motherboard and upgrades from XP to Vista and still insists "I've made no changes," there can occur very serious changes in hardware that users might not realize have happened. Especially with aging laptops it might be the case that your hard disk or disk controller or motherboard is failing in a way that has blotted out key sectors on your disk, much as degenerative dementia blots out parts of human brain function. In computers as in humans this can be a destructive result of the aging process.
• You are utilizing virtualization software and have forgotten that each such virtual machine is a true machine: bring those machines into existence and out and you are indeed changing your computer.

Unfortunately, given the inherent complexities of the latter three scenarios it is up to you to debug what is going on with your system. If you are sure the hardware is not failing, the best course to recover from chaos caused by unknown programs or viruses is to re-format the hard disk, to reload Windows from original Microsoft media (to avoid infected backups), to install industrial-grade antivirus software with current virus definitions and then finally to restore data and programs from backup media after careful screening for viruses. You can then re-install Manifold and use one of your remaining activations to activate the new Manifold installation.

I have several Manifold System licenses and serial numbers. Can I freely mix Activation keys? No. Each Activation key will work only with the serial number and System ID that was used to obtain it from the Manifold Activation key server.

I entered my serial number and Activation key into the Activation dialog but Manifold will not launch or the serial number and key are not accepted. Check the obvious first:
• Did you correctly enter the serial number and the Activation key? Use Copy and Paste to avoid typographic errors. Be sure you are copying the entire string and pasting correctly without any errors. Do not make any changes, such as removing hyphens or changing from upper to lower case. Use the original serial number, not the version masked off with XXX characters at the end.
• If you own multiple Manifold licenses, have you mixed up a serial number for one license with an Activation key that was fetched using a different serial number? Activation keys only work with the original serial number, not the version masked off with XXX characters at the end.
• Did you obtain the Activation key from the Manifold Activation key web page? If you obtained your serial number or Activation key from a third party it may not be valid.
• Have 72 hours passed since the time the Activation key was created by the Manifold Activation key web page? If so, the Activation key has expired. You will need to get another Activation key.
• Are you trying to use a serial number and Activation key issued for an extension (such as Business Tools, Geocoding Tools or Surface Tools) to activate Manifold System itself, or vice versa are you trying to use a Manifold System serial number to activate an extension?
• Is the system date reasonably accurate? Serial numbers and Activation keys encode the date they were issued and will not work with system dates that are obviously wrong such as a system date before the date of issue of the serial number or Activation key.
• Are you using a nonstandard program that replaces or modifies the standard Windows interface so that Copy and Paste operations are modified?
• Are you running any programs in background other than standard Windows processes and services? When running the Activation dialog, make sure no other programs are running in background except standard Microsoft Windows processes and services. Do not run anti-virus checkers, uninstallers, registry "protectors," network checkers, special administrative protection programs or other such programs. These may interfere with the activation and permanent installation process.

I lost my serial number. Can you help? Possibly yes. However, a key recovery fee will be charged for this service. See the products listing page in the main manifold.net web site for prices charged for Key Recovery services, if available, as well as terms and conditions for such recovery.
Can I transfer a Manifold System license from one computer to another? During the preliminary installation period you can install Manifold System on any computer system using only the serial number. After the preliminary installation period has expired, a license transfer will require normally require reinstalling Manifold System and using up an activation.

Can I upgrade my computer system without re-installing Manifold? As long as you don’t make major changes in hardware or software, probably yes. Simple upgrades such as adding a new graphics card will normally not require a re-installation of Manifold. Some changes in hard disk structure will require re-installation of Manifold and the use of an activation. The only way to know for sure is to try. Accordingly, it is wise to do any major upgrades during the period of preliminary installation.

Can I upgrade Windows without re-installing Manifold? In most cases, no. Installing a new version of Windows will usually require a re-installation of Manifold and use of an activation.

Can I remove and re-install Manifold without a new activation? Yes. Even if you remove Manifold, on the next installation Manifold will know that you have already activated Manifold on this system so it will not require a new activation, provided that you have not made any changes to your system in between the uninstall and the re-install that would wipe out activation.

Can I install a new Manifold serial number without a new activation? No. Installing a new serial number requires a new activation. Activation works only for a limited time for the serial number and system ID used to fetch the key. For example, if you have installed Professional Edition and you would like to switch to Enterprise Edition you will need an Enterprise Edition serial number and a fresh activation for that new Enterprise Edition serial number. See the Help - Activate topic for discussion and an example.

Can I install a Manifold Update or Service Pack without a new Activation key? Yes. You can uninstall the previous Manifold installation and then install the update, which also may be known as a Service Pack (SP), without requiring a new activation. The system will “remember” the previous activation even though Manifold was uninstalled. Of course, this assumes you have made no other changes to your system that would require a new activation in between uninstalling the previous Manifold installation and then installing the update. You cannot, for example, uninstall the previous Manifold installation, install a new version of Windows, and then expect to install an update without requiring an activation.

Can I use a serial number from one edition of Manifold with a newer edition of Manifold? No. Each new major release of Manifold requires a new serial number.

What is a “substantial” change in Windows that requires a new Activation key? Any re-installation of Windows or upgrade to a different edition of Windows will probably require a new activation. Other major configuration changes (such as Registry configuration or rollback to a pre-Manifold restore point) may require a new activation.

Why do Activation keys have a limited life of 72 hours? Activation keys have a limited life of 72 hours from the time they are issued in order to discourage attempts to clone installations in violation of the license and to otherwise discourage software piracy.

Why are serial numbers limited to five activations? To discourage piracy. The preliminary installation period allows time to sort out the initial Manifold installation before the first activation is used. After that, four additional activations will allow users to re-install should Manifold ever need to be re-installed in the future. Five activations are plenty to allow users to deal with changing system configurations for a single license, but they are not so many that people can afford to be careless with them or to give them away to enable piracy. It’s not a terrible limit since Manifold gives out a free activation every year and provides additional activations at absurdly low prices.

I upgrade my computer system frequently and I need more than five activations. What do I do? Take advantage of free activations, low cost additional activations or additional activations through upgrades. If taking advantage of all that is not enough, you must buy additional Manifold licenses to get the activations you need. If you are a member of an organization that is operating a Manifold System License Server you can avoid burning up activations by taking advantage of the License Server.

What is to prevent me from giving my serial number to four friends who each then obtain an activation? If you are willing to break the law, nothing prevents you from doing so except your own self-interest. From a legal perspective, that is a crime and a violation of the License agreement. Engaging in software piracy risks criminal and civil penalties. As a practical matter, it is risky to do because your friends may use up the activations you need yourself and they probably aren’t going to bother to call you to coordinate their use of Manifold with you so that the same serial number is not used at the same time on different installations. If you ever need an activation
to re-install Manifold (if case your hard disk blows up or if you must re-install Windows) you might not have an activation remaining. In addition, whenever you disclose your serial number to anyone else you take the risk that they will not keep it secret. The sort of person who trades serial numbers to commit software piracy is often the sort of person who cannot resist passing on the serial number to yet another friend. Before you know it, your license rights could be used up. Your serial number might even be published on a hacker web site and your serial number could be cancelled by manifold.net, permanently denying you the ability to use Manifold System. Last but not least, if you disclose your serial number to someone else they might use it to upgrade to some wonderful new Manifold release before you can do so. They'll have the wonderful new release and you won't be able to get it, because your serial number will have been revoked when it was traded in on an upgrade. It is easy to avoid such problems by keeping your serial number secret.

Can I use the same Activation key to install Manifold to more than one system during the 72-hour period the key is enabled? No, because each Activation key is fetched by the System ID that identifies that request.

What happens if I run the same license on more than one machine at the same time? Do not violate the License by using the same license on more than one machine at the same time. Manifold System can detect when other instances of the same license are running on machines that are accessible to it. If you use the same license on multiple machines at one time, sooner or later Manifold will automatically cancel that serial number and will cease functioning. To discourage software pirates who attempt to find and disable such protective measures, this auto-detection function might be set to a very long trigger time: it could be weeks or months of simultaneous usage before Manifold decides that "enough is enough" and permanently stops functioning.

My organization needs to use many Manifold licenses. How should I install them? Use the Manifold System License Server product to centrally manage your licenses.

I acquired my Manifold license from a reseller and have not received a serial number. Can manifold.net help me? No. If you acquired your Manifold license from a third party and have not received your serial number you must contact the third party to resolve the difficulty. manifold.net will not intervene in a dispute between you and a third party. manifold.net does not enter into any special agreements with any dealers or distributors to represent the company so any third party who offers to transfer a Manifold license to you is acting on their own.

What happens if my serial number is revoked? If your serial number is cancelled or revoked you will not be able to obtain activations with that serial number. In addition, sooner or later your Manifold System license will cease operating.

How might my serial number be revoked? The most frequent reason is that it has been traded in to get an upgrade. For example, if you upgrade a Professional license to an Enterprise license the Professional serial number will revoke the Enterprise license. Likewise, upgrading a 32-bit serial number to a 64-bit serial number will revoke the 32-bit serial number. Serial numbers may also be revoked for a variety of business or legal reasons: non-payment, such as failure to pay a purchase order or a charge-back on a credit card transaction; automatic cancellation when Manifold itself detects multiple simultaneous usage of the same serial number on a different machine; cancellation for violation of the EULA, for example, cancellation by manifold.net if your serial number appears in a public forum such as an Internet posting.

I want to acquire a Manifold license and run it forever, or at least for decades. Can that be done? Sure. Once installed a Manifold license will run as long as that computer continues to operate in sufficiently unmodified form to not require re-activation. If you are running exactly the same computer hardware without failures you can indeed run Manifold for decades. In addition, the five activations provided for a serial number enable re-installation and re-activation within a reasonable period of time; however, re-activation of old or obsolete licenses in the case of machines that fail on a regular basis may not continue to be supported once future versions of Manifold are released. Although manifold.net may voluntarily support use of activation keys for outdated versions of Manifold for some reasonable period of time (such as two or three years) licensees should not expect such support. You should therefore budget for acquiring updates from time to time to enable Manifold installation on newer computer systems or on a repaired computer system if over the years hardware failures require re-installation.

Serial Numbers not Received by Email

When an order is processed for Manifold System the serial number for that license will be sent out by email to the email address given in the order. If you have not received your serial number by email from manifold.net check your email system for obvious errors. Typical errors include:

- You did not provide an email address when you acquired your Manifold license. This happens occasionally when purchasing groups neglect to provide an email address on a purchase order. If so, manifold.net is waiting to hear from you to learn what email address you wish to use.
• You acquired your license from a third party, such as a reseller or previous licensee. In that case, you must contact that third party for any questions about serial numbers as manifold.net is not part of your business deal with that third party. Note that manifold.net does not enter into any agency or reseller agreements or "authorize" any dealers, so if you acquired a Manifold license from some third party that third party is in no way an agent or representative of manifold.net.

• A nonexistent email address (typographical error or otherwise invalid email address) was supplied with your order. Tech tip: check the invoice and the packing slip sent to you in hard copy with the product - it will contain the email address you provided. Check that printed email address carefully for typos to be sure it is the email address you are using.

• You have forgotten to check the inbox for the email address you supplied. Tech Tip: If your Outlook application is set up to check multiple addresses, make sure the address you provided is one that is being checked for inbound mail.

• You accidentally deleted the email without realizing it contained your serial number. Tech Tip: Take a look in your Deleted Items folder to see if it is there.

• Your mailbox is over quota and so your ISP is rejecting inbound emails.

• You have not used the email address provided to manifold.net for a long time and so your email provider has suspended it as inactive.

• The email address supplied to manifold.net has been forwarded to a different email address and the forwarding is not working.

• You have changed your email system recently and it is not working (relaying not permitted, etc.) in the new configuration.

• You have created a new email address just for business dealings with manifold.net and it is not working. Tech tip: Don't use a previously-unused email address that's not been verified to work. manifold.net won't spam you or share your email address with anyone else.

• You or your ISP have installed anti-spam technology that is too aggressive and is blocking the emails being sent to you by manifold.net. Check any "spam" or "bulk mail" folders into which your spam filter may be placing emails prior to deletion.

• Your or your ISP have installed a virus or other filter that is too aggressive and is blocking manifold.net emails because it thinks they contain scripts or executables (they don't).

• You use Outlook and have applied some mail-handling rules that are deleting inbound messages from manifold.net or moving them to a folder that you do not view.

• You or your ISP have installed a "blacklist" filter that blocks emails originating from IP addresses used by Manifold email servers or email servers run by Manifold's ISP if such IP addresses are reported by a centralized listing organization to be spam servers. [Note that such filters are pointless since spammers routinely forge the origin of their spam to make it look like it comes from legitimate IP addresses.]

• You have placed manifold.net as a domain on your anti-spam whitelist but have not allowed email from other domains used by our email servers or from dedicated IP addresses.

• You or your ISP have installed spam-blocking technology that requires the sender to reply to a confirmation email or to visit some web site and enter a code. The Online Store and other Manifold servers that send out serial number emails are machines that cannot respond to return emails requiring confirmation.

• Someone else, such as a centralized IT department, runs your email system and is being a completely uncooperative bureaucrat about telling you what stupid tricks they use to prevent you from getting legitimate email. Tech tip: get your own free HTML-based email account and use that to bypass the IT department.

Everyone on the planet is totally annoyed about spam so it is natural that people want to use anti-spam technology to help get rid of it. Unfortunately, no anti-spam technology is perfect so you may have to exercise some manual supervision over your anti-spam strategy to make sure it does not filter out legitimate emails from manifold.net that you want to receive.

If one of the problems above is in play, that can lead to a very frustrating experience with communications. For example, a customer sometimes without realizing it will have massive spam filters in place that prevent any communications from manifold.net getting through. The customer does not receive the serial number email and then when the customer writes to sales@manifold.net to complain, he or she does not get any reply. The customer sends repeated emails, each of which gets angrier and angrier. The problem, of course, is that if the customer has some spam filter that rejects emails from manifold.net then no matter how often manifold.net replies the customer will not get those replies.

The solution to the above problem is to use some temporary, HTML-based, free email address from hotmail.com, yahoo.com or Google's gmail to correspond with manifold.net.
The worst situation is when someone for ego or broken logic reasons is not willing to consider the possibility that something about his or her email system is preventing email from manifold.net from being delivered. There are two classic cases: the first case being the person who thinks that because they receive plenty of emails from other people there cannot be anything wrong with their email setup. This is the "broken logic" problem because obviously you won't know about the many emails you don't get if you never get them.

The second typical case happens when the person having email reception problems is the person who has set up the email system and has ego problems considering that something might not be quite right. Some people, for example, seem to have a knack for setting up their own email servers that are configured with all sorts of tricky arrangements that don't correctly. No doubt it has something to do with inadequate fiber in their diets. It is difficult to solve a problem when such self-administered email gurus are unwilling to consider they may have made an error.

If communications are the problem, keep it technical: get a free email account and use that to work around whatever problem exists.

**A Note on Copy Protection**

Professionally produced counterfeit or pirated software is widely available in some countries. However, the use of such software poses risks for the unwary, including the risk that the pirated software package is actually transmitting your location to police authorities. See the Beware of Counterfeit Software essay for a discussion of some of the little-known risks of using such software.

Some folks are outraged that a software company might utilize anti-piracy measures such as copy protection. Against the chance that some such people might be interested in the practical reasons why copy protection is employed, or that a sensible person may be called upon to calm down an angry colleague, we provide the following information:

With over 40% of the software in the US in commercial settings being used in violation of licenses and well over 95% of software in some countries being pirated copies, it's just not cost effective to create and offer software like Manifold at very low cost if copy protection methods are not employed. If you are very angry about that and disagree, that's OK: do not license Manifold software. There are always alternatives that may better fit your preferences for licensing, although none of them will have the extensive capabilities and ultimately lowest costs of Manifold.

For those folks who do choose to license Manifold, the license is not an unlimited one nor is it constructed to allow all possible users to run Manifold until the end of time no matter what happens. Providing that capability would be very expensive and would ruin the low cost that makes Manifold affordable to so many users. Instead, the license and copy protection mechanisms try to strike a balance that makes reasonable use possible at as low a cost as possible for as many users as possible. It's a formula that has brought real GIS capabilities to more users than ever before.

Experienced users will not get hung up on the idea of "buying" a license that will run "forever," as attempting to do that will simply raise costs in most cases. It's not even a realistic proposition, as people who attempt it usually discover.

Computer hardware and software technology change so rapidly that installations more than a few years old become antiques. They become difficult to keep running because replacement software and hardware parts are not available and because newer software requires newer technology. You know exactly what we mean if you've ever tried to find RAID drivers for that old box made in 2001 you are trying to press into service today as a web server running the most recent edition of Windows. As a practical matter the great majority of software versions should not be expected to have a lifetime of more than three to five years at the most.

Manifold issues new editions more frequently than that and provides a low cost upgrade path that makes it easy to get newer versions and additional activations for a modest fee. Some folks don't like the idea that to run Manifold "forever" they have to, in effect, "rent" a license by paying modest fees for upgrades every now and then. However, that is a necessary consequence of evolving systems: because Manifold is not freeware newer versions will not be free. The costs of creating new editions are very high and manifold.net is not willing to create such new editions without compensation. Likewise, the price we all pay for endemic software piracy is the limitation on usage and increased costs associated with copy protection mechanisms.

The current system is the least cost way manifold.net has found after years of experience to assure copy protection, to provide a low initial cost and to keep the costs of upgrades as low as possible. At the time of this writing, the manifold.net system has resulted in GIS software costs that are far lower than any other full-featured GIS software, in some cases over forty times less expensive than competitors. That's an amazing
accomplishment that could only have been financed by assuring, through copy protection using serial numbers and activations, that everyone has paid their fair share.

It's true that there will always be atypical cases where some users need more activations or some users might really intend to keep antique configurations running for many years. Sometimes that happens when people have unique usage patterns, such as using laptop computers in very harsh environments where the computers need to be replaced with unusual frequency no matter how careful the user may be. Such users will have higher costs and it is only fair that they should bear their own higher expenses for their special needs.

In other cases some people live disorderly, inattentive, unplanned lives and as a result will end up wasting activation keys. Some people acquire a Manifold license and then go on vacation for 30 days, so that when they return they no longer have a "free" period of 30 days before activation is required in which they can sort out any machine configuration problems. Some people don't bother to read even simplified instructions and so waste keys despite numerous warnings of lack of Administrator status or other admonitions. Some people don't bother to use anti-virus software or don't bother to familiarize themselves with Windows administration, so they find themselves unnecessarily burning through activations as they try to deal with the damage inflicted on machines by viruses or by imprudent machine management. Such people will encounter higher costs to license Manifold because they will have to acquire additional licenses to replace the activations they have wasted.

In most cases, even such unusual needs are not a problem because additional activations are so easy to get either for free or for very low costs. Between free yearly keys, very low cost additional keys and inexpensive keys from upgrades, it is really very difficult to run out of free or low cost keys. But some people have such extraordinarily unusual needs they might do so.

The least cost way manifold.net has found to service the above extraordinarily unusual needs, without forcing other users to bear the costs of those unusual needs or, in some cases, gross negligence, is to simply offer new licenses at the regular price. Of the many systems one can imagine to validate and deal with unusual situations that turns out to be the least costly method. The goal served by that method is to keep the cost of GIS as low as possible for as many people as possible.

Note: Users who want to run old machine configurations for many years should be aware that future editions of Manifold might not run on very old configurations, so if that is the intention it would be wise order a few extra Manifold licenses of the vintage of interest while such licenses are still available. We think a better idea is to simply get with the program and upgrade machine hardware, Windows systems and Manifold editions on a regular basis so that you are always working on the latest, super-cool new stuff the computer industry delivers. You don't have to upgrade every month, but trying to run antique hardware for 10 or 20 years is just not productive.

A related topic that may be useful in some cases, such as understanding what happens to a license if former partners squabble over a Manifold serial number when dissolving a partnership, is the Maintaining Your Manifold License topic.

See Also

Command Line Options
Help - Activate
Help - Activate Extension
Installing and Activating a Manifold Extension
Maintaining Your Manifold License
Manifold System License Server
Runtime Licenses
Installing and Activating a Manifold Extension

This topic supplements the information in the Activation Keys and Serial Numbers topic. Please read that topic before continuing with this topic.

Installation

The Manifold System product line contains optional extensions, such as the Surface Tools and Geocoding Tools and Business Tools packages, which extend the functionality of Manifold by adding new commands and capabilities. These optional extensions are not delivered separately; instead, they are part of the code within every Manifold System installation and simply must be "unlocked" for functionality.

If you have licensed one or more of the three optional extensions you will receive a serial number that identifies the license for that extension. That serial number may be used to turn on that extension and to activate it for permanent use. The serial number and activation process used is similar to that used for Manifold System itself.

If you have licensed an extension, do not search for some special installation package or special setup program. If you have installed Manifold System you have already "installed" the code necessary for that extension. All you need do is use the serial number for that extension to activate the extension.

• If you have installed Manifold System you are ready to activate a Business Tools, Geocoding Tools or Surface Tools extension using the serial number for that extension that was sent to you.
• If you have not installed Manifold System you cannot activate or use an extension. Extension products require previous installation of Manifold System.
• If you do not have the serial number for the desired extension package, you will not be able to activate that extension.
• Serial numbers for extensions must match the release level of the Manifold System installation. For example, to activate Surface Tools on a Release 8.00 Manifold System installation you must have a 8.00 Surface Tools license.

Activating an Extension

The Help - Activate Extension dialog is used to provide a serial number and to activate a desired extension. When Help - Activate Extension is launched, the system will close any open projects and raise the Extension Activation dialog to allow entry of a serial number or of a serial number and an Activation key. The system will then shut down Manifold so that Manifold may be re-launched with fresh activation information. When Manifold is re-launched it will automatically be reconfigured so that the activated extension will be fully enabled.

There is no need to specify which extension is being activated, since each serial number for an extension is specific to the extension that it activates. For example, if you license the Surface Tools package from manifold.net you will receive a Surface Tools serial number. Entering that serial number into the Extension Activation dialog will automatically turn on Surface Tools.

Just like the serial number used to enable Manifold, serial numbers for extensions typically allow a temporary period of usage, normally 30 days, without requiring activation. This preliminary period allows time for users in remote areas to acquire an Activation key while still being able to use the extension. While an extension is operating on serial number alone the Help - About dialog will report how many days are left on that serial number before activation is required.

Note that some product bundles licensed by manifold.net may use a single serial number both for Manifold and also at the same time for any Manifold extensions. For example, the Universal Edition Manifold System product configuration provides a single serial number that turns on Enterprise Edition features and also turns on Business Tools, Geocoding Tools and Surface Tools features as well.

If you license a Manifold System edition other than Universal Edition and then also license Business Tools, Geocoding Tools and Surface Tools, you will receive four serial numbers: one for Manifold System and one for each extension. To install and activate all of these products you will have to use the Manifold System serial number to activate Manifold.

You will then have to use the serial number for each extension to activate that extension. You will end up going through the activation process four times, once for Manifold System and then once again for each of the two extensions. For this reason, licensing Universal Edition is a real convenience for managers and users because it allows use of a single activation procedure to install and activate Manifold System as well as all extensions.
**Example: Activating Geocoding Tools**

Activating a Business Tools or Surface Tools license uses exactly the same procedure as that given below using the Geocoding Tools extension as an example.

If you license the Geocoding Tools extension you will receive a serial number for your Geocoding Tools license by email. If you license more than one Manifold product in the same order all of the serial numbers for all the products in that order will be in a single email, clearly identified by product for each product licensed.

2. Launch Manifold System. Close any projects that may be open.
3. Choose Help - Activate Extension
4. In the Extension Activation dialog enter the Geocoding Tools serial number. Enter it exactly as it was issued to you by manifold.net. Do not change upper case to lower case. Do not replace hyphens with space characters or make any other changes to the serial number. If desired, you may also enter an Activation key for the serial number as well. Press Accept.
5. Manifold will then exit. When re-launched, the Geocoding Tools package will be enabled.
6. Using the serial number alone the Geocoding Tools package may be run for 30 days from the date the serial number was issued. After that, an Activation Key must be fetched and provided to the Help - Activate Extension dialog together with the serial number. Once an Activation Key and a serial number have been provided to the Help - Activate Extension dialog the extension will be permanently installed. Don't forget to login as Administrator when permanently activating the Geocoding Tools extension.

See the Activation Keys and Serial Numbers topic for information on how to get an Activation key. If your computer is attached to Internet you can get an Activation key automatically from the Extension Activation dialog by pressing the Get Activation Key via the Web button in the dialog. If your computer is not connected to Internet you can get an Activation key manually as discussed in the Activation Keys and Serial Numbers topic.

**Caution:** Just like your main Manifold System serial number, you only get five activations per serial number for an extension, so be just as careful with such activations as you would be with your Manifold System license.

**Command Line Activation**

Extensions may activated via a command line as set forth in the Activation Keys and Serial Numbers topic.

**Troubleshooting**

Visit the Support page on the manifold.net web site for links to the Activation Guide and Activation Troubleshooting pages.

The most common problems reported by users when activating extensions are:

- Administrator login not used. You must login as the Administrator in Windows 7, Windows Vista, XP, 2000, Server 2003 and Server 2008. See the Vista Admin Login link in the Support page if you think you are logged in as Administrator but in fact are not.
- Mixing up serial numbers. Except in the case of Universal Edition, you cannot use a main Manifold System serial number (such as a serial number for Manifold System Professional Edition) to turn on an extension. You must use the serial number for the extension. Likewise, you cannot use a serial number for an extension to turn on a main Manifold System license. For example, a Surface Tools serial number cannot be used to activate a main Manifold System Professional Edition license.
- Inaccurate system date. Your system date must be accurate to install an extension. The extension's serial number codes the date when it was issued and when it requires an Activation key and Manifold will not allow the serial number to be used if you attempt to use it outside the allowed dates.

**See Also**
Runtime Licenses

Manifold System licenses may be acquired as runtime licenses as an alternative to regular licenses. A runtime license provides all Manifold functions and capabilities except for those accessed through the console GUI (graphical user interface). Runtime licenses are used when deploying an IMS application to a web server or when deploying an application that calls Manifold capabilities programmatically through the API (applications program interface).

Note that regular licenses also provide console GUI functioning as well as IMS (Internet Map Server) and API capability. Therefore, when developing an application that calls Manifold one normally does so using a regular license since it provides both the console GUI and also IMS and API functionality. This gives the developer the ability to learn Manifold interactively using the console GUI as well as the ability to try out runtime functionality for an IMS or other application.

When the application has been developed and debugged and is ready to deploy, runtime licenses may be used to lower the cost of deployment.

The serial number provided when activating Manifold determines if Manifold will run in regular mode or in runtime mode. The serial number further identifies which Manifold capabilities are enabled. For example, providing a serial number for a Manifold Professional Edition Runtime license during activation will cause Manifold to subsequently execute only in runtime mode with Professional Edition features available either in IMS or otherwise through the API. Providing a Manifold Universal Edition Runtime serial number during activation will enable subsequent runtime use of Universal Edition features such as Business Tools features.

Once Manifold is activated using a runtime serial number then Manifold will execute only in runtime mode and no dialogs will be raised to remind the user when the initial 30 day period of installation for a serial number has expired. It is therefore wise to always activate a runtime license of Manifold using both a serial number and an Activation key so that when the initial 30 day installation period expires no one will be left wondering why it is that Manifold capabilities are no longer available on the machine. Attempting to launch Manifold interactively on a machine when a runtime license has been installed will display an error message.

Keep in mind when activating Manifold with a runtime serial number that once Manifold is so activated on a particular machine the runtime activation status will be remembered even if Manifold is subsequently uninstalled and then reinstalled. Thereafter, unless the activation status of Manifold is changed, Manifold will execute only in runtime mode. This means that changing the activation status from a runtime license to a non-runtime license will require using command line activation, since launching manifold.exe interactively will simply raise an error message.

See Also

Activation Keys and Serial Numbers
Command Line Options
Manifold System License Server

All Manifold licenses are licensed and copy protected using one of two mechanisms:

- **Standalone licenses**, authorized using a Serial number and Activation keys that locks each license to a single computer system. This is similar to how Microsoft Word and other such applications are typically licensed. Manifold licenses acquired in small quantities or acquired for use on portable devices use this mechanism. If you have not acquired a Manifold System License Server product you should turn to the Activation Keys and Serial Numbers topic for a discussion of serial numbers and activation for your license.

- **Floating licenses**, authorized using a Manifold License Server. Manifold licenses acquired in larger quantities will frequently use this mechanism, which requires licensing of the Manifold System License Server product.

The rest of this topic discusses the Manifold System License Server.

**License Server Overview**

The **Manifold System License Server** product gives organizations the ability to maintain an inventory of floating Manifold licenses that may be used by any client computer that can connect to the License Server, all without needing to use serial numbers and Activation keys for individual client licenses.

License Server offers organizations several key benefits:

- **Fast and easy deployment** - IT managers can deploy Manifold using standard automated deployment tools with no need to individually activate each machine.
- **Security for license rights** - Serial numbers need not be handled at individual client machines, thus providing greater protection against theft of license rights. The master serial number authorizing License Server rights need not be shared outside the trusted group of IT managers.
- **Fewer licenses required** - Fewer licenses need be procured because only the maximum number of licenses that will be used at any given time need be acquired. Licenses need not be acquired for individual machines just against the chance that someone will need to use Manifold on that machine.
- **More efficient use of different Manifold editions** - License servers make it easy to have a range of Manifold editions available within the organization so that if a user normally works with Enterprise Edition but occasionally requires Universal Edition the higher edition can also be run. This reduces the number of higher end licenses that need be acquired if most users normally work with lower level licenses.
- **Simplified machine upgrades** - Because authorization is not stored on client machines it is not sensitive to upgrades in machine hardware or software. Client machines may be replaced, reconfigured or upgraded as much as desired without burning up a limited stock of client activations.
- **Reduced administrative overhead** - IT administrators need keep track of only one serial number per License Server no matter how many licenses have been acquired. That’s far easier than keeping track of individual serial numbers and activations per client machine.
- **Easier on End Users** - Personnel using Manifold on client machines don’t need to know anything about activation and can simply launch Manifold and get to work. End users cannot cause havoc on client machines by misusing activation commands from the Manifold menu.
- **Web service connectivity** - The Manifold License Server runs as an HTTP web service to which client machines connect for authentication. Most organizations have already configured internal firewalls and proxies to allow users to browse internal websites on their intranet or even to browse external websites on Internet. Running License Server as a web service, therefore, makes it likely that it will work “out of the box” in most organizations or, at worst, require only a few well understood, easy to apply and easy to test adjustments to proxy settings or network configuration.

While License Server is great, it does have some constraints on usage:

- **Costly for small installations** - Running a License Server installation requires acquisition of the Manifold System License Server product plus acquisition of the desired number of licenses. Because of the cost of a License Server product plus the cost of the desired licenses is greater than acquisition...
of small numbers of non-License Server licenses, individual users and very small organizations will usually not acquire License Server.

- **Local web server required** - A License Server requires a machine on your local, private network that can function as a local web server running Microsoft IIS to host the License Server and to authenticate floating licenses via your local, private network. This machine may be used for other purposes but cannot be used to run Manifold other than as a License Server. Only one License Server can be installed on each machine. Although License Server can be successfully used to authenticate floating licenses for Manifold installations on virtual machines, such installations involve many nuances and uncertainties arising from the virtualization software in use; therefore, they are not supported by manifold.net.

- **One edition per License Server** - A License Server can only host licenses for a single Manifold Edition. If we need to distribute floating licenses for both Enterprise Edition and Universal Edition within our organization, we will have to acquire and operate two License Servers, one to serve Enterprise Edition licenses and one to serve Universal Edition licenses. The two License Servers will require two machines.

- **Extensions via Universal Edition** - A Manifold License Server can be used to authenticate only main Manifold System licenses and not optional extensions. If we require an extension such as Business Tools, Geocoding Tools or Surface Tools we must run a License Server that serves Universal Edition licenses. Universal Edition includes all three extensions within the main Manifold System edition. Runtime licenses are not available as floating licenses in License Server.

- **Connection to License Server** - License Server supports use of floating licenses only on machines that can connect every few minutes to the License Server using a private network. Notebook computers and other machines that do not have network connectivity with the License Server must use regular, static licenses. Most organizations that employ License Server for desktop machines will also acquire a number of individually-licensed Manifold System licenses for users of portable computers.

- **IT / Manifold expertise required** - While the installation and management of License Server does not require any unusual expertise, it does require sufficient technical skills to set up a simple web page and to resolve any proxy, firewall or other network connectivity issues between client machines and the License Server. Administrators should also have solid Manifold experience with clear understanding of and experience in installation and activation of Manifold licenses.

- **Not a way to lend or lease licenses** - Using License Server does not allow you to bypass the lending or other restrictions in the Manifold EULA. License Server provides more convenient and economically efficient licensing within one organization - it is not a way to share, loan out or lease Manifold licenses between different organizations or different individuals.

**Acquisition**

To license Manifold System License Server, contact Manifold Sales at sales@manifold.net or by telephone. Acquisition requires licensing the License Server product and at the same time licensing the number of Manifold System licenses for the edition desired to be served by the License Server. See the manifold.net web site for detailed sales information for License Server.

**Delivery**

The Manifold System License Server is simply a Manifold System Edition like any other. Download the standard installation file from the Product Downloads and install it by installing Manifold System like any other edition. Like any other Manifold edition, License Server is configured to be License Server by the serial number that is provided to authenticate the License Server configuration. Using a License Server serial number in the Manifold Activation dialog will configure that installation to function as a License Server. Also, download the licenseservermisc.zip file from the Product Downloads page for the .asp files mentioned below.

Once Manifold has been configured by the serial number to run as a License Server it will no longer run as an interactive GIS nor will it provide IMS or be accessible via the Manifold API for GIS functioning. The configuration will only function as a non-interactive License Server that can be utilized exclusively through the web service created by installation (see below).

The serial number issued when License Server is acquired will encode the Manifold System edition supported by that License Server as well as the number of Manifold System licenses authorized by that server.

**Activation**

Read the Activation Keys and Serial Numbers topic for details on the Manifold System activation process. Each License Server installation will use that process to activate the License Server on the target server machine.
Administrators installing License Server are assumed to be experts who are completely familiar with the Manifold System activation process described in the above topic.

The activation process is just like that used for individual licenses with the following exceptions:

- **Server key required** - Ordinary licenses use a **Serial number** and a **System ID** to fetch an **Activation key**, which is then used in the Activation dialog along with the Serial number and System ID to activate the installation. In addition to those three keys the License Server activation process also uses a fourth key, the **Server key**, which is provided by the Manifold activation web site. This fourth key is required for an .ini initialization text file placed on the License Server machine.

- **Manual activation required** - Ordinary licenses can use the Activation dialog to fetch an Activation key through the web. License Server installations must use manual activation. The person activating the License Server installation must visit the Manifold activation web site and use the Serial number and System ID to fetch an Activation key and a Server key. When the Manifold activation web site gets a valid License Server serial number, it knows that a Server key must be provided in addition to an Activation key.

- **No 30 day initial period** - Manifold System editions other than License Server can run for 30 days from the date the serial number is issued before activation is required. License Server serial numbers do not have this 30 day initial period. Instead, a License Server requires permanent installation with both a serial number and an Activation key for functioning.

In a nutshell, the process of installing and activating a License Server installation is as follows:

- Install and launch Manifold on the server. This launches the Activation dialog, which displays a System ID.
- Copy the System ID from the Activation dialog.
- Visit the Manifold activation web site. Enter the System ID from the Activation dialog. Also enter your Serial number, copied from the serial number email sent for that License Server license. Use Copy and Paste at all times to avoid typos.
- The activation web site generates an Activation key and a Server key. Copy the Server key to a safe place, like a text file that is immediately saved, on the License Server machine. Copy the Activation key and the Serial number into the Activation dialog and press Accept.
- Create a Manifold.ini text file using the Server key as described below and place it in the Manifold installation folder on the License Server machine.
- Copy the LicenseServer.asp file from the licenseservermisc.zip download file into a folder in the C:\InetPub\wwwroot hierarchy on the License Server so it can be accessed as a web service. Make it a web application in IIS.

On client machines, there is no activation necessary. Simply install Manifold on each client machine and then place a Manifold.ini text file created as described below in the Manifold installation folder on each client machine.

Like all Manifold system serial numbers, a License Server serial number is good for five activations and five activations only. Take care not to waste activations as it may be costly to get additional activation should you run out. A License Server serial number is licensed for use on a single server machine. Do not install the same License Server serial number on more than one machine, as you may lose your license rights completely if the same serial number is used to enable operation of more than one License Server at the same time.

Like all Manifold keys, the Activation key and Server key issued by the Activation web page are fetched by the specific Serial number and System ID used to obtain them.

Unlike other serial numbers for other Manifold editions, a License Server serial number does not allow a period of preliminary installation using the serial number alone. Instead, License Server must always be permanently installed using both a serial number and an Activation key. This is a safety measure because License Server normally controls very many licenses, potentially many thousands of Manifold licenses, and so cannot be placed at risk of a sudden stop if a preliminary 30 day period of operation expires because the IT manager running the installation forgot to permanently activate the installation.

License Server administrators should therefore be familiar with Manifold hardware and software requirements as well as with Manifold activation. Choose a stable, reliable machine for your server that won't have to go through hardware reconfiguration changes and which otherwise fulfills the general requirements (see the main Manifold web site) for running Manifold software.
Upgrading

To increase the number of licenses authorized by a License Server, please see the sales information at the manifold.net website or contact Manifold Sales at sales@manifold.net. Increasing the number of licenses will be accomplished by trading in your existing License Server serial number and being issued a replacement serial number that encodes a greater number of licenses.

Once the new serial number is in hand, it can be used to upgrade the License Server installation following the instructions that will be provided by manifold.net. You will need to launch License Server again and provide a new serial number to the Activation dialog to increase the number of authorized licenses.

Note that since License Server is, in effect, a runtime installation, to get the activation dialog you will have to use command line activation. The easiest way of doing this is to supply an invalid serial number in a command line activation to force Manifold to pop open the Activation dialog. For example, using the command line switch:

```
manifold.exe /activate:dialog
```

...will pop open the Activation dialog since "dialog" is not a valid serial number. See the Command Line Options topic and the Activation Keys and Serial Numbers topic for details.

Very important: You can only increase the number of licenses authorized by a License Server if that License Server product is the current Manifold System product being offered. When new versions of Manifold are published, after a few months the old version is discontinued and is no longer available. License Server licensees who may want to increase their licenses should make sure to transition their License Server installation to any new version of Manifold System that may be published, for example, by taking advantage of any limited time discounted upgrade offers for transitions to the new version.

Installation

Manifold License server may be installed on Windows installations equipped with .NET Framework 2.0 and running Internet Information Services 5.1 or more recent IIS. The following examples use ordinary internet URLs to make it clear how URLs are used. Keep in mind that your License Server must be accessed through a private network and not placed on the open Internet.

To install Manifold License Server:

1. Log in using a user account with Administrator privileges.

2. Locate the serial number email that was sent to you with your License Server serial number. Open the email so that you can copy the serial number when it is needed.

3. Install Manifold System if not yet installed.

4. Launch Manifold. When Manifold is installed onto a new machine, the Activation dialog will pop up when Manifold is launched. If using an existing Manifold installation Manifold will launch in the usual way. In that case, choose the Help - Activate command to open the Activation dialog.

   - Note that the Activation dialog has a System ID box that is filled with a System ID value to be used when fetching the Activation key. The Activation dialog also has boxes for a Serial number and an Activation key.
   - Copy the serial number from the serial numbers email and paste it into the Serial number box in the Activation dialog.

5. Launch a web browser and go to the Manifold activation page at:

   http://www.manifold.net/activation

   - Copy the serial number from the serial numbers email and paste it into the Serial number box in the web page.
Activation and Licensing

- Copy the System ID from the Activation dialog and paste it into the System ID box in the web page.
- Click the Create Activation Key button on the web page to create an Activation key and a Server key. These keys are fetched by the Serial number and System ID provided.

6. Copy the Activation key from the web page into the Activation dialog. The Activation dialog will now have the Serial number, System ID, and Activation key boxes filled. Press Accept. Manifold will open a box complaining it cannot run in interactive mode. Press OK. This message box shows that the License Server activation has occurred, since License Server cannot run interactively but can only run as a web service.

7. Create a new text file with the following contents:

```
[Activation]
ServerKey=XXXB4E2F4-6BAE9A-D54DE1E061E99BC7FA6FC58C82B4765
```

Instead of the fake key shown above that begins with "XXX" use the Server key generated by the Manifold activation web page for your serial number and system ID. Use Copy and Paste to transfer the Server key from the web page into the text file to avoid typographical errors.

- Save the text file into the Manifold installation folder (by default, C:/Program Files/Manifold System) as Manifold.ini. This file tells the License Server what Server key to use when License Server starts.

8. Create a new web site by copying the LicenseServer.asp file found in the licenseservermisc.zip download file into a suitable folder within the C:/inetpub/wwwroot hierarchy. Be sure to configure the web site to be a web application in the IIS configuration console.

9. Create a new text file with the following contents:

```
[Activation]
ServerUrl=http://myserver.com/manifold/LicenseServer.asp
```

Replace the fake URL above with the actual URL for the LicenseServer.asp file.

Save the text file into a temporary folder as Manifold.ini. You will need to copy this Manifold.ini file into the Manifold installation folder (by default, C:/Program Files/Manifold System) on each client machine that will run Manifold and will seek license authentication from the License Server. This file tells client instances of Manifold how to contact the License Server for permission to run.

If client machines will be connecting to the license server machine via a web proxy, also include connection info within the Manifold.ini file that is deployed to client machines using the following template:

```
[Activation]
ServerUrl=http://myserver.com/manifold/LicenseServer.asp
Proxy=<proxy server address>
ProxyUser=<proxy server user name>
ProxyPassword=<proxy server password>
```

If desired, specify the timeout for web requests to the license server machine by adding the following entry to the Manifold.ini file:

```
RequestTimeout=<timeout in milliseconds>
```

Note: in the above template examples do not actually employ the "<>" characters, which are used in the above only to indicate information that is to be provided. For example, if we would like a 30 millisecond timeout we would use:

```
RequestTimeout=30
```
10. On each client machine that is to be served by the License Server, install Manifold System but do not yet launch it. Copy the Manifold.ini file created in the previous step into the Manifold installation folder (by default, C:/Program Files/Manifold System) on that machine. You can now launch Manifold and it will automatically use the information in the Manifold.ini file to connect to License Server and get permission to launch. If all the licenses are not in use, Manifold will launch in the normal way. If all licenses are in use or if the license server cannot be contacted, an error message will result.

**Operation**

**On the server:** When IIS runs the LicenseServer.asp page, the License Server first checks the Manifold.ini file for a valid Server key. The internal cryptographic engine uses that key to determine that License Server is indeed authorized to work on this machine to serve the authorized number of licenses of the authorized Manifold edition. It then proceeds to service client requests up to the authorized number of licenses.

**On the client:** When Manifold launches it checks to see if a Manifold.ini file is present in the Manifold installation folder. If so, it knows to connect to a License Server for license authentication and follows the URL given to ask for a license. Manifold accepts whatever license is authorized by the License Server URL.

You can test the operation of the License Server by launching Manifold on a client machine that you know has network connectivity to the License Server web service. Manifold should launch normally. Choose the Help - About dialog to see what Manifold Edition is reported in use and confirm it corresponds to the Manifold Edition authorized by the license server.

To see how many licenses are being served, use a small Licenseservercheck.asp file as discussed below.

**Client Connection Logic**

Client instances of Manifold configured to connect to a License Server send it their System ID during startup, which the server uses to distinguish different client instances from each other. The server tells each client system whether or not the instance is allowed to run. The identification of client instances by system IDs allows running any number of Manifold instances on the same client system, all served by a single license slot.

If a client instance fails to obtain permission to run from the License Server upon startup (due all available licenses being in use or the server being unreachable), it refuses to run. If a client instance successfully obtains permission to run, it will launch Manifold and will thereafter contact the License Server every 5 minutes to maintain its license slot. If the instance fails to contact the License Server for more than 15 minutes, the License Server will consider that instance to be abandoned and makes its license slot available for use by other instances.

If the original instance tries to contact the License Server after a disconnect of greater than 15 minutes, it will find its license slot used up and will try to acquire another license slot. If the license server does not have any other license slots available, the instance will suspend operation and will wait until one or more of the currently running instances of Manifold give up their license slots. During the wait, the instance will display a modal Execution Suspended dialog, which is by default configured to try reconnecting to the server automatically every minute.

**Troubleshooting**

Assuming IIS is running correctly and the LicenseServer.asp file has been installed as a web application within the C:\InetPub\wwwroot folder hierarchy, the usual problem is a network configuration (firewall, etc.) that prevents a client machine from connecting to the License Server.

An easy way to check this is to put a small index.htm file comprising a simple web page in the same folder as the LicenseServer.asp file and then to attempt to launch that web page from a browser on the client machine. For example, if trying

http://myserver.com/manifold/index.htm

in your browser can't get a web page connection then you know

http://myserver.com/manifold/LicenseServer.asp
won't work in Manifold.ini either.

Next, create a small .asp test file as described in the Create a Simple ASP File section of the troubleshooting IMS topic. That will verify that ASP is turned on in your web server as well.

Don’t forget to check the obvious:

- Is IIS running and web pages (see above) on the License Server can be accessed from client machines?
- Is Windows firewall or other software firewall, if used, configured to allow outbound web server HTTP connectivity?
- Is IP Filtering enabled for the network connection? If so, make sure port 80 is allowed to enable HTTP traffic.
- Are any hardware firewalls or port filters in place between the server and client? Make sure they are configured to pass HTTP.
- Is a proxy server used? If Manifold fails to launch and a proxy is in use, make sure the values of the ProxyXxx entries in the Manifold.ini file are correct and the value of the RequestTimeout entry, if used, is not too low.

See the Support page on the manifold.net website for any additional notes that may be published on License Server operation. Because License Server troubleshooting is almost always network debugging that has nothing to do with Manifold, technical support for License Server is handled using developer level support incidents.

Security

Administrators who work with License Server in large local area networks will often want to establish security measures to ensure that their floating licenses are not used up by unauthorized personnel. Security may also be required to ensure that use of License Server is kept within the licensed organization, a requirement of the Manifold license for License Server.

Because License Server is implemented as an HTML server it is easy to apply the usual security measures employed within Windows networks and IIS web sites to control access to a resource, in this case access to the HTML server that hosts License Server.

Two examples of such security measures include:

- **Proxy Server**: A proxy server could be set up on a bastion machine through which connections to the HTML server must pass. The password to access the proxy server could change from time to time to discourage unauthorized use.
- **VPN**: Even better, the HTML server could be on an intranet accessible only via VPN where credentialed access to the VPN is required.

Other general advice for security:

- Keep confidential the serial number for the license server.
- Maintain physical security on the license server machine.
- Maintain security between the license server and its clients (usually done with two-way authentication and some kind of encryption, as with a VPN).

Tech Tips

You can switch back and forth between local license authorization and License Server authorization on client machines.

When Manifold launches it first looks for a Manifold.ini file containing a ServerURL entry. If present, Manifold launches using the license authorized by the server cited in the Manifold.ini file. If no Manifold.ini file is present then Manifold checks for local activation status. You can use this startup logic to go back and forth between License Server authorization and local activation status by simply providing or removing a Manifold.ini file within the Manifold installation folder.
Suppose you install and activate Manifold Professional Edition using a Professional Edition serial number on a particular system, say, a laptop computer. If no Manifold.ini file appears in the Manifold installation folder Manifold will launch as Professional edition.

When the laptop is connected to the organizational network and can reach a License server, if you want to switch to using Enterprise Edition from a License Server you can copy a Manifold.ini file that points to the License Server into the Manifold installation folder. When Manifold launches it will see the Manifold.ini file and the ServerURL entry it contains and use that to launch as Enterprise Edition.

If you then remove that Manifold.ini file (or, as most people do, simply rename it to Manifold.txt or some other name), when Manifold launches it will not find a Manifold.ini and will launch Manifold using the local activation status as Professional Edition.

If you have several different Manifold License Servers available you can switch between authorized editions by simply changing the Manifold.ini file to point to whatever server you want. For example, if you have a License Server that authorizes Enterprise Edition licenses and another License Server that authorizes Professional Edition licenses you could keep two files in your Manifold installation folder, one called Manifold.ent and one called Manifold.pro, that contain the text for a Manifold.ini file that points to either an Enterprise edition License Server or a Professional edition License Server. Depending on which server you want to use you can copy either the Manifold.ent or the Manifold.pro file to the Manifold.ini file.

IT managers who want to automate the above can write a simple batch file that copies the desired ServerURL entry into a Manifold.ini file, allowing users to choose which Manifold edition they want to use when a selection of License Servers is available.

Checking the number of licenses in use is easy. Create a small page called Licenseservercheck.asp with the following code. Install the page on the machine hosting the License Server. Users can then navigate to that page from their clients to see how many licenses are currently being served. This page is provided in the licenseservermisc.zip download. The code is trivial:

```html
<%@ enablesessionstate=false language=javascript %>

Response.ContentType = "text/plain";
Response.Expires = 0;

// report current number of outstanding sessions
var svr = Server.CreateObject("Manifold.LicenseServer");
var res = "Current Sessions: " + svr.CurrentSessions; svr = null;

Response.Write(res);
%
```

See Also

Command Line Options
Activation Keys and Serial Numbers
Maintaining Your Manifold License

Manifold System is licensed, not sold. That means your right to use Manifold System is subject to the terms and conditions set forth in the End-User License Agreement (EULA) to which you agreed when acquiring and installing the software. The license to use Manifold is not unlimited: there are significant limitations in the license, such as access to a limited number of activations and limitations on transfer that must be observed. Anyone who is in possession of a Manifold license has agreed to the terms and conditions set forth in the EULA. If you have not agreed to those terms and conditions, you are not licensed to use Manifold products.

In addition to the rights extended by the EULA manifold.net normally also voluntarily provides additional benefits to licensees, such as the opportunity to take advantage of certain discounted upgrade offers, sample usage of technical support products through two free standard technical support incidents, possible access to additional activation keys (either free or paid) and other benefits. Such benefits may be withdrawn or modified at any time without notice.

This topic explains in an informal way how manifold.net administers licenses so that you can take maximum advantage of your license. This topic does not replace or amend or alter in any way the EULA that sets forth the terms and conditions of your license. If there is any conflict in any way between this topic and the EULA, the EULA takes precedence and is the sole and complete authoritative statement of your license for Manifold System.

Routine Maintenance

By default, a Manifold product license requires no maintenance except retaining the original serial number email that issues the license and taking care to observe the terms and conditions of the EULA. For example, install the license on one computer and don't give out your serial number to other people.

There are no annual fees or compulsory support fees required. The default Manifold System standalone license is a fully paid-up license. Once installed on a computer it can run forever if that computer is not reconfigured in a way that destroys the activation of that license (significant disk changes, re-installation of Windows, etc.). Multiple activations are provided for each serial number so the license can be re-activated should significant reconfiguration occur on a computer or should the license be moved to a new computer. See the Activation Keys and Serial Numbers topic for details.

Very important: Direct support and free services from manifold.net are available only to the original licensee who is in possession of the original serial number email and is writing from the email address to which that serial number email message was sent. Manifold.net does not support third parties, such as customers of resellers who acquired a Manifold license from manifold.net and then provided that license to the reseller's customer. In such cases, the reseller must support his or her customers.

As is set forth in the EULA, it is OK to transfer your license to someone else; however, because manifold.net has no control over what is told to such a recipient or whether all Manifold materials provided to the original licensee have been provided to that recipient, manifold.net will not support that recipient with any free services. In general, manifold.net only provides free services, such as installation or activation support, to the original licensee.

Proof of License

As noted in the Activation Keys and Serial Numbers topic, a Manifold license is authorized by the specific serial number for that license. Anyone who has the serial number for a license can use that serial number to install the product and to get activations for that license. Conversely, anyone who does not have the serial number cannot utilize the license.

A Manifold license is controlled by possession of the serial number that authorizes that license. Licensees are identified by possession of the Manifold System serial number associated with the license. To comply with privacy regulations and other requirements, in any transaction with someone who claims to be a Manifold licensee manifold.net must know that the person is, in fact, the licensee.

The sole way a licensee can identify themselves is by producing the serial number for the license. If you do not have the serial number, you are not the licensee. If you lose your serial number, you have lost your license and, if you cannot use one of the optional key recovery services if such may be offered, you must procure another license to replace the one you lost. Further, any free services for activation or installation support from manifold.net are available only to the original licensee who is in possession of the original serial number email and is writing from the email address to which that serial number email was sent.
Lost Serial Numbers

Licensees want to have a functioning installation when they first install Manifold, so they usually pay reasonable attention to the installation instructions in the serial number email. Typical text used in serial number emails to advise users in the strongest possible terms not to lose serial numbers is as follows:

PLEASE SAVE YOUR SERIAL NUMBERS

DO NOT LOSE THESE SERIAL NUMBERS. IF YOU LOSE THESE SERIAL NUMBERS, THEY CANNOT BE RECOVERED.

If you lose these serial numbers, you will lose the ability to permanently install or to use the products they authorize, you will lose the ability to re-install your products or to move them to a new machine should you ever need to do so in the future and you will lose any ability to take advantage of future upgrade offers that require a Manifold serial number.

Do not expect that after using a serial number once you can safely lose it. Serial numbers are stored in Manifold products in encrypted form and cannot be recovered from the software installation: you will need these original serial numbers, so DO NOT LOSE THESE SERIAL NUMBERS.

PRINT OUT A COPY OF THIS EMAIL AND SAVE IT IN A SAFE PLACE.

Users who have installed their own Manifold licenses no doubt remember reading the above and have taken care to save their serial numbers. Likewise, most organizations likewise are careful to assign trustworthy individuals to manage valuable assets such as serial numbers. For example, an IT department that installs licenses on behalf of users will normally employ diligent employees who will take note of the above advice and who will make sure to save serial numbers in the IT department's records.

There are cases where a Manifold license has been acquired within an organization and information about the serial number, who ordered the procurement, how the license was obtained and how the license was utilized are hopelessly lost. For example, if Manifold was acquired by a person no longer employed by the organization then it could be the license has been irretrievably lost. In some cases organizations don't even know if they've acquired a Manifold license or if they have whether it was an employee's personal license, provided by a contractor or consultant, obtained through a reseller or through some other channel.

A Manifold serial number is like a prepaid telephone or cash card - if lost, it cannot normally be recovered. It is not tied to a particular user or machine or even a download. If you lose it, you should consider it gone. Under some circumstances for some types of licenses it might be possible, for a fee, for manifold.net to attempt to recover a serial number if the customer has detailed information regarding the original transaction. However, even in the best of circumstances such recovery may be unlikely. Therefore, optional key recovery service products to attempt a recovery might not be offered given the low chances of success. If offered, such optional key recovery service products will require a significant fee.

Key recovery services may be offered for a fee that have a good chance of recovering lost serial numbers if the information required for recovery exists. If your organization has lost your serial numbers and does not have enough information to use any optional key recovery services that may be offered, then your licenses have been irretrievably lost and you must acquire replacement licenses at the regular price. Replacement licenses are not discounted because it costs more to determine if a discount is warranted than the cost of fresh licenses.

Contacting manifold.net

For manifold.net to recognize you as a Manifold licensee when contacting the company on matters relating to your license you should include the serial number for your license. manifold.net will always take possession of the serial number as proof positive the correspondent is the licensee. Note however, that possession of a serial number does not entitle you to any services, such as tech support or other services, which are not part of the license. Such services are normally provided using support tokens or other mechanisms, some of which may be limited to the original licensee. Free services are provided only to the original licensee who is in possession of the original serial number email and is writing from the email address to which that serial number email was sent. See the Support pages on the manifold.net web site for additional information.

Running out of Activations
Manifold is not licensed for endless usage no matter how many times you need to re-install it. Each serial number can be used for only a limited number of activations. Through sensible use, prudent computer administration and by taking advantage of free additional keys, low cost additional keys or low-cost upgrades that may be offered from time to time it is fairly easy for virtually all users to continue using Manifold without running out of activations at a very low cost per year or at no cost at all. See the Activation Keys and Serial Numbers topic for details.

See also the Support pages on the manifold.net web site for any optional services (either free or paid) that may provide a route for additional activations for a serial number.

**Payment does not Confer a License**

The licensee is the person who has the serial number. That can be, and often is, a different person than the person who pays for the license. A very common situation is where one party is the credit card holder paying for the transaction while a different person is the user who receives the serial number email and who actually uses the product.

Occasionally in the course of disputes such as partnership dissolutions the person who paid for a license will be angry that manifold.net does not regard that person as a "co-licensee" or will not take instructions from him or her as the licensee. However, if you do not have the serial number you are not the licensee and you have no rights in the license at all even if it is you who paid for the license.

The situation is analogous to what happens if you buy a car for cash and then give it to someone else so the car is registered in that other person's name. As far as the car registry is concerned, if your name is not on the title to the car it doesn't matter that you paid for it. The person whose name is on the title is the owner of the car. Although car titles may be encumbered by lien holders, there is no such thing as an encumbrance on a Manifold license. Whoever has the serial number controls that license.

If you pay for a Manifold license, keep in mind that the user receiving the product and serial number will be the licensee. If you don't trust the licensee and wish to retain some degree of control over the license, provide your email address as the initial email address to which the serial number is sent. You can then forward the serial number to the ultimate user. However, in the meantime you can keep a copy of the serial number so that you too have physical possession of the serial number.

**Transferring a License**

The EULA allows you to transfer your license once to someone else. A license may only be transferred once, so the person to whom you transfer your license cannot subsequently transfer their license to someone else.

To transfer your license, provide the serial number to the new licensee. When transferring a license, you must also transfer all Manifold materials, such as product installation download files or serial numbers, to the new licensee. You must also delete Manifold System from any computer on which you have installed it, since if the same serial number is active on more than one computer a license violation sooner or later will be detected and the license will be revoked.

A transfer is a permanent change of licensee. You cannot lend a license to someone else by temporarily transferring your license to them and then having them transfer the license back to you. Once you transfer your license to someone else, you lose all rights in that license.

On occasion manifold.net will receive emails from people who say that a Manifold licensee has transferred their license to them and that manifold.net should update the license record accordingly and send them the required serial number. Such emails have no effect because only the licensee who possesses the serial number can transfer the serial number to a new licensee. If you think you have received a Manifold license from the original licensee but have not received the serial number for that license, you must take that up with the original licensee. If they have not provided you with the serial number for that license, you are not a Manifold licensee.

If you are acquiring a Manifold license from a third party, you must be able to trust that third party not to cheat you - it is a "buyer beware" situation. You must trust that the serial number provided to you is a valid serial number that matches the software provided and is the edition desired. For example, if you want to acquire an Enterprise license make sure that you are getting an Enterprise serial number and not a Professional serial number. You must trust that the serial number still has a sufficient number of activations remaining from the original five so that you can use the software as you desire. For example, if the serial number provided has only one activation left you will not be able to install Manifold both on a desktop machine and on a portable computer. Finally, you must trust that the person providing Manifold to you has indeed uninstalled Manifold from their computers and will not provide the serial number to anyone else, to be sure that your serial number will neither be revoked nor used up without your knowledge.
Activation and Support

Please make sure to read the Activation Keys and Serial Numbers topic carefully to understand the activation mechanism used to copy-protect Manifold before acquiring a license from any third party. To check the status of a serial number, please visit the Support page on the manifold.net web site for links to the status-checking page.

Very Important: Note that under no circumstances will manifold.net provide activation assistance for anyone but the original licensee writing from the original email address used for the license. If whoever procured the license you wish to use tells you otherwise, you are being misled.

Arguments between Partners

There are many circumstances in which the ownership of licenses may be in dispute. For example, partners who acquired Manifold for a mutual business may break up their partnership and subsequently argue over who gets the Manifold license. Partnerships are dissolved, couples divorce, businesses go bankrupt or are restructured, employees quit without making note of serial numbers, resellers might take money for Manifold licenses and not deliver the product and many other circumstances occur in which licensing rights become muddled in the eyes of participants.

In all such cases, manifold.net does not get involved in any private disputes. Under no circumstances will manifold.net intervene. Whoever has possession of the serial number controls the license. If more than one party knows the serial number and there is a dispute between those parties that is a problem between the parties which they must resolve on their own.

Disputes with Resellers

Manifold.net is a direct Internet sales company that does not designate resellers or authorize distributors. No person or company is authorized by manifold.net to in any way make any representations on behalf of manifold.net or to in any way act on behalf of manifold.net. No person or company is authorized by manifold.net to change the terms and conditions of licensing or any other business policy of manifold.net.

If you acquire a Manifold license from a reseller or other distributor, that reseller has simply acquired Manifold just as anyone else can do and has no special authorization from manifold.net to act in any way on behalf of manifold.net. If you deal with such a person you are engaged in a business deal that has no connection with manifold.net. If that person lied to you and told you they were an agent of manifold.net or otherwise specially "authorized" to represent manifold.net that is a problem between you and the person who lied to you.

The situation is just like that if you buy some used electronic equipment from a private seller on eBay. If that private seller lied to you about the nature of the equipment or neglected to give you various missing parts required to operate that equipment your dispute is with that private seller and not with the original manufacturer of the electronic equipment.

There are many consultants, VARs and resellers of various kinds in the GIS industry who are high-integrity people and who choose to use Manifold as part of the solution they offer. Such people can often be the source of outstanding value and service to customers, but customers must always be aware that in such cases they are dealing with a third party and not with manifold.net or any agent of manifold.net. Choose a reseller whose integrity you can rely upon.

Because the originating reseller is Manifold's customer, not the person who buys from the reseller, people who buy from that reseller must be able to trust the reseller not to cause that license to be revoked by financial or other cheats. For example, if the reseller procured a license using a credit card, resell the license to the reseller's customer and then a few months later the reseller charges back the credit card transaction that license will be revoked. Even if the chargeback is not successful (manifold.net disputes all chargebacks) the license will not be useable for future discount offers or for other transactions requiring the use of a credit card because it will have been permanently associated with an attempted credit card fraud.

Regrettably, a premium brand like Manifold will always attract the attentions of dishonest people who will falsely claim association with the brand in order to steal some of the brand's prestige for themselves. Just as some crooks will falsely claim to be "authorized" Rolex dealers when they are not, some people will claim to be "authorized" Manifold dealers when they are not. As always in business, customers should be wary of imposters and vendors of either counterfeit goods or counterfeit relationships.
In some cases the problem might not be outright fraud but just simple incompetence. Some middlemen are not able or willing to follow even the simplest of instructions or invest even the bare minimum of self-education but are nonetheless interested in using the booming market interest in Manifold to turn a quick dollar for themselves. Instead of facilitating software acquisition and installation such middlemen might cause chaos and frustration for their customers. Customers should not involve middlemen who do not have the expertise and willingness to support them and who do not bring true value added to the transaction.

Example 1

John Doe and Harry Doe are brothers working together in a partnership. John Doe orders Manifold System and receives the serial number. He gives the serial number to Harry. One year later the partnership dissolves acrimoniously and the partners verbally agree that Harry should get the Manifold license since he is the more technical person.

Harry Doe sends an email to manifold.net with the serial number for the license and directs manifold.net to change the license record to make Harry the licensee of record. manifold.net replies that whoever has possession of the serial number is the licensee and that the company does not update license records in that way.

A few weeks later, John Doe decides he doesn't like the verbal deal he made with Harry. Unknown to Harry, John emails manifold.net to get the serial number (he has mislaid it) and is informed that if he does not have the serial number, manifold.net cannot provide it to him.

John is upset, but when he cools down he sends all sorts of long letters and copies of partnership agreements to show that the partnership has dissolved and he, being the one who really paid for Manifold in the first place, should be the licensee of record. manifold.net politely replies that none of that matters as the company does not take sides in any dispute. Anyone with the serial number can control the license and someone who does not have the serial number is out of luck. The serial number is not tied to any user or any one machine. Anyone who has it can use it. Really.

After a while, John asks that if he finds his serial number can he order a transfer of the license back to him. manifold.net politely responds that whoever has the serial number as a practical matter can use it. John is not happy, but finally decides to honor the deal he made with his brother. He hires a new GIS guy for his new business and gets a fresh copy of Manifold.

Example 2

Jane Doe has acquired Manifold from a reseller who has assured her (falsely) that he is the exclusive Manifold authorized reseller in her geographic area. When she downloads and installs Manifold and launches the product, she discovers she needs a serial number. She calls her reseller who tells her to contact manifold.net to get a serial number.

For the sake of this example we don't know why the reseller tells Jane to contact manifold.net for the serial number. It could be that the reseller is dishonest and has already installed the serial number on the reseller's machine. Or, it could simply be that the reseller is negligent and hasn't bothered to pay attention to forwarding the serial number email to Jane.

Jane emails manifold.net requesting a serial number, providing a copy of her sales receipt from the reseller. manifold.net politely responds that Jane is not a Manifold licensee and thus manifold.net cannot discuss the status of any license with her. Jane is also advised to contact whoever provided Manifold to her to make sure that the original licensee transfers the license to her and also provides her with the serial number.

Jane becomes angry with manifold.net because she thinks the company owes her something since she acquired Manifold through an authorized reseller and so she writes a lengthy, angry email.

manifold.net politely responds that there is no such thing as an authorized Manifold reseller and no such thing as exclusive geographic territories and that if someone so represented themselves they did not tell her the truth. Jane is informed that manifold.net does not get involved in any dispute between her and any third party. The letter from manifold.net to Jane concludes by citing the manifold.net web pages that state for the record manifold.net does not authorize any agents or resellers.

In the above case, Jane's best recourse is to work the issue with the reseller, getting the reseller to give her the serial number for the license (which most likely was emailed to some email address provided by the reseller). Jane can then use the serial number as she sees fit.
Note that **manifold.net** is not at all involved in the above transaction since the company does **not** authorize any resellers or have any special contractual arrangements to allow any third party to represent the company. **manifold.net** is a direct Internet sales company and does not use distributors or resellers. However, anyone can purchase from the company and then subsequently resell and transfer a license to someone else as allowed by the EULA.

The situation is no different than if someone walks into a computer store, purchases a Microsoft product, and then later misrepresents himself to other people as an authorized Microsoft reseller, sells them the Microsoft product he bought in the store and fails to provide essential information required for use. In that case, Microsoft is not involved at all in that transaction since Microsoft has no control over what people do with the product once they buy it in a store, as anyone can do.

**Example 3**

Harry Doe goes to work for a large company as a temporary contract employee and suggests the company begin using Manifold System. He orders a copy of Manifold using a personal credit card (he is later reimbursed for the expense by the company) using his home address and gives a *hotmail.com* email address since he has not yet been issued a company email address. When the serial number email arrives he downloads and installs Manifold on a machine at work and begins using it.

A few months later the temporary project is completed and Harry is hired by another company. He carefully hands over his serial number email and other information to his replacement, explaining the need for the replacement not to lose the serial number and to review **manifold.net** tech support procedures and how to use the serial number. But the replacement is a new guy who is somewhat careless. He doesn't bother to review the information or save the serial number. A year later the computer on which Manifold was installed crashes and Manifold needs to be re-installed and re-activated, but no one can find the serial number for the license or even remember how the company came to be in possession of Manifold.

The company sends **manifold.net** an email explaining the circumstances, asking **manifold.net** to provide a list of all licenses acquired on behalf of the company and requesting a transfer of any licenses to a current employee. **manifold.net** politely replies that if the customer no longer has a serial number they may have an opportunity to use a key recovery service, for a fee, but otherwise any serial numbers are indeed lost.

In this case the company should have taken better care to manage its intellectual property assets, such as license rights, to avoid losing valuable assets during a change of personnel. The company's best option is to contact Harry and to ask him if he still has a copy of the serial number or if he remembers how the company came to possess a copy of Manifold.

Eventually, they find someone who remembers Harry and they send him a note asking him about the serial number. Unlike his careless replacement, Harry is a good guy and very organized and has kept a copy of the serial number exactly because he suspected his replacement would be negligent. He sends in the serial number.

However, suppose no one could remember how the company got Manifold or perhaps Harry was not a good guy or left the company under negative circumstances and thus was disinclined to do any favors for a company that no longer employs him. In that case the company probably has no practical recourse other than to pay for a new license or to take a chance using whatever key recovery service might be available.

**License Terminations**

If you do not follow the terms and conditions of the EULA **manifold.net** will revoke your license to use Manifold System. Licenses will also be revoked for non-payment. For example, if you chargeback a credit card transaction with **manifold.net** you will lose any Manifold licenses procured in that transaction and you will permanently lose any access to transactions requiring the use of a credit card, such as discounted upgrades to future products, even for any products you procured that were not involved in the chargeback.

Note that a Manifold System license cannot be loaned or shared, nor can it be installed on more than one machine except that in addition to installation on one desktop machine it may be installed on a portable computer used principally by you so long as that Manifold license on the portable machine is not used at the same time as the one on the desktop machine. In a nutshell, you cannot allow your Manifold license to be used on multiple machines. If you do, sooner or later the license violation will be discovered and your license will be revoked.

Do not lend to or share Manifold System with other parties. You may transfer your license on a one-time basis, but note that you must provide your serial number and all materials without retaining any functioning installations of Manifold System. If you acquire a Manifold license and then provide it to some third party without transferring
all use as required by the EULA, it is only a matter of time before manifold.net becomes aware that more than one person is using the same Manifold license, causing the license to be revoked.

Do not publish serial numbers or Activation keys or other tokens in any public forum. Doing so will revoke the license.

Summary

The bottom line is that it is very easy to keep your Manifold license in good shape: don't make illegal copies of Manifold, don't install one license on multiple machines, don't loan out or share your Manifold license and don't post your keys on the web. Make sure to save a copy of your original serial number email in a safe place. If you do decide to transfer your license to someone else, do not retain your original installation. You cannot continue running Manifold if you transfer your license to someone else. You must uninstall your original license, delete all Manifold materials in your possession and transfer both the serial number and all materials to the new licensee so both you and the recipient comply with the EULA.

If you acquire a Manifold license from someone else it is up to you to make sure the serial number you receive still retains sufficient activations for you to get the use you desire from that license.

Tech tip: You can check the status of any serial number by visiting the Serial Number Status page that is a link available in the Support page and in many other pages on the manifold.net web site. By entering a serial number into the Serial Number Status page you can see if it is a valid serial number, what Manifold edition it authorizes and how many activations have been used.

See Also

License
Activation Keys and Serial Numbers
1. **GRANT OF LICENSE.** This Section of the EULA describes Your general rights to install and use the Software Product. The license rights described in this Section are subject to all other terms and conditions of this EULA.

- **General License Grant to Install and Use Software Product.** For each serial number key issued to you by Manifold You may install and use one copy of the Software Product on a single computer, device, workstation, terminal, or other digital electronic or analog device ("Device"). You may make a second copy of the Software Product and install it on a portable Device for the exclusive use of the person who is the primary user of the first copy of the Software Product provided that the Software is not used on both Devices simultaneously. A license for the Software Product may not be shared.

- **Alternative License Grant for Storage/Network Use.** As an alternative to the rights granted in the previous section, You may install a copy of the Software Product on one storage Device, such as a network server, and allow individuals within Your business or enterprise to access and use the Software Product from other Devices over a private network, provided that You acquire and dedicate a license for the storage Device upon which the Software Product is installed and each separate Device from which the Software Product is accessed and used, except however that any usage of map server functions from a separate Device exclusively via an interactive Internet browser that is generally marketed and available to the public shall not require a separate license for each such separate Device. A license for the Software Product may not be used concurrently on different Devices.

- **General License Grant to Install and Use Subscription Product.** The following licensing terms apply to You instead of the license grants in the previous two paragraphs if You licensed a subscription-based Software Product (a "Subscription Product"). You may install one copy of the Subscription Product on a single Device and use the Subscription Product for the term of Your subscription. You may also exercise the additional license rights described in the paragraphs below, but only for the term of Your subscription. The initial subscription period begins on the date You first activate Your copy of the Subscription Product and ends three hundred and sixty five (365) days thereafter. You cannot use the Subscription Product after Your subscription expires unless You renew or extend Your subscription. By renewing or extending Your subscription, You will be entitled to continue using the Subscription Product for a specified period of time beyond the date when Your previous subscription would have otherwise ended. All the terms and conditions of this EULA will continue to apply to Your use of the Subscription Product during any subsequent renewal periods unless otherwise specified. After the expiration of Your subscription, You can continue to open, view and print any documents You created with the Subscription Product.

- **Additional License Grant for Data Sets.** The Software Product may include certain digital maps or digital cartographic data sets, other data sets, documents, photographs, clip art, animations, sounds, music and video clips (together "Data Sets"). If so, the following terms describe Your rights to the Data Sets:
  - Except as specified in the next Section, You may use, copy and modify the Data Sets and distribute copies of the Data Sets, along with Your modifications, as part of Your software product(s) and service(s), including Your web site(s).
  - You are not licensed to do any of the following:
2. DESCRIPTION OF OTHER RIGHTS AND LIMITATIONS.

- **Copy Protection.** The Software Product may include copy protection technology to prevent the unauthorized copying of the Software Product. By way of example and not limitation such copy protection technology may require original media for use of the Software Product on the Device, may require dongles or other hardware mechanisms, may require the use of key codes, may restrict the Software Product installation to a single Device, may detect via a network if this EULA is being violated, may require Administrator login privileges for installation or may limit the number of re-installations allowed. It is illegal to make unauthorized copies of the Software Product or to circumvent any copy protection technology included in the Software Product.

- **Authentication Keys.** For correct function, this Software Product may require a serial number key, product activation key or other authentication keys ("Authentication Keys"). Such keys may be limited in number to limit the number of installations possible or for other copy protection purposes. If You lose any Authentication Keys Manifold may charge You a fee to replace or re-issue lost Authentication Keys if, in Manifold's sole and exclusive choice, Manifold chooses to offer such replacement or re-issuance services to You. Any Authentication Keys or other identifying information authenticating this license are licensed, confidential information and may not under any circumstances be disclosed to any third party except pursuant to a permanent transfer of this Software Product. You agree to allow Software Product to communicate with authentication servers from time to time without notice to authenticate the validity of any Authentication Keys required by Software Product to function. You agree to keep any such Authentication Keys strictly confidential (including any serial number keys issued to You by Manifold).

- **Not for Resale Software.** If the Software Product is labeled "Not For Resale" or "NFR," then, notwithstanding other sections of this EULA, You use of the Software Product is limited to use for demonstration, test, or evaluation purposes and You may not resell, or otherwise transfer for value, the Software Product.

- **Limitations on Reverse Engineering, Decompilation, and Disassembly.** You may not reverse engineer, decompile, or disassemble the Software Product, except and only to the extent that such activity is expressly permitted by applicable law notwithstanding this limitation.

- **Separation of Component Parts.** The Software Product is licensed as a single product. Its component parts may not be separated for use on more than one Device unless expressly permitted by this EULA.

- **Trademarks.** This EULA does not grant You any rights in connection with any trademarks or service marks of Manifold. You agree not to infringe any trademarks or service marks of Manifold.

- **Preservation of Notices.** Any customization or configuration of Software Product by You must retain all Manifold notices setting forth trademarks, copyrights, Software Product origin, licensing information, contact information for Manifold, limitations on usage or references to any Manifold sites in at least as prominent a fashion as in the original configuration of Software Product.

- **No rental, leasing or commercial hosting.** You may not rent, lease, lend or provide commercial hosting services to third parties with the Software Product, except that commercial hosting services consisting of Internet publication of .map file projects using the map server embedded in the Software Product is permitted, provided that You acquire a license for Software Product for each server used for such hosting service.

- **Indemnification.** You agree to defend, indemnify, and hold harmless Manifold, its officers, directors, shareholders, parents, agents, contractors, affiliates, subsidiaries, joint ventures, licensors, owners and employees, from any and all damages, costs, claims, taxes, levies, losses, expenses, fees, attorneys fees, liabilities or any other costs or expenses of any kind, including payment of same by You, that in any way arise from or result from the acquisition, licensing, possession, transfer or use of Software Product by You.
3. UPGRADES.

- **Support Services.** Manifold may provide You with support services related to the Software Product ("Support Services"). Use of Support Services is governed by the Manifold policies and programs described in the user manual, in "online" documentation, or in other Manifold-provided materials. Any supplemental software code provided to You as part of the Support Services are considered part of the Software Product and subject to the terms and conditions of this EULA. You acknowledge and agree that Manifold may use technical information You provide to Manifold as part of the Support Services for its business purposes, including for product support and development. Manifold will not utilize such technical information in a form that personally identifies You.

- **Software Transfer.** Except as specified in this Section, the initial licensee of the Software Product may make a one-time permanent transfer of this EULA and Software Product only directly to an end user. This transfer must include all of the Software Product (including all component parts, the media and printed materials, any upgrades, any Authentication Keys, this EULA, and, if applicable, the Certificate of Authenticity). Such transfer may not be by way of consignment or any other indirect transfer. The transferee of such one-time transfer must agree to comply with the terms of this EULA, including the obligation not to further transfer this EULA and Software Product. Subscription products are non-transferable. No transfers are permitted for Academic Licenses.

- **Termination.** Without prejudice to any other rights, Manifold may terminate this EULA and/or order Software Product to cease functioning if You fail to comply with the terms and conditions of this EULA. This Software Product may automatically terminate this EULA and cease functioning if it detects a licensing violation such as, by way of example and not limitation, unauthorized simultaneous shared concurrent usage of this license on different computers. If this EULA is terminated by Manifold You must destroy all copies of the Software Product and all of its component parts.

4. INTELLECTUAL PROPERTY RIGHTS. All title and intellectual property rights in and to the Software Product (including but not limited to any trademarks or service marks, digital maps or digital cartographic data sets, data sets, documents, images, photographs, animations, video, audio, music, text, and "applets" incorporated into the Software Product), the accompanying printed materials, and any copies of the Software Product are owned by Manifold or its suppliers. All title and intellectual property rights in and to the content that is not contained in the Software Product, the accompanying printed materials, and any copies of the Software Product are owned by content owners and may be protected by applicable copyright or other intellectual property laws and treaties. This EULA grants You no rights to use such content. If this Software Product contains documentation that is provided in electronic form, you may print one copy of such electronic documentation. You may not copy the printed materials accompanying the Software Product. You agree not to infringe any intellectual property rights in and to the Software Product.

5. BACKUP COPY. After installation of one copy of the Software Product pursuant to this EULA, you may keep the original media on which the Software Product was provided by Manifold solely for backup or archival purposes. If the original media is required to use the Software Product on the Device, you may make one copy of the Software Product solely for backup or archival purposes. Except as expressly provided in this EULA, you may not otherwise make copies of the Software Product or the printed materials accompanying the Software Product.

6. U.S. GOVERNMENT LICENSE RIGHTS. All Software Product provided to the U.S. Government pursuant to solicitations issued on or after December 1, 1995 is provided with the commercial license rights and restrictions described elsewhere herein. Software Product is not licensed for provision to the U.S. Government pursuant to solicitations issued prior to December 1, 1995.

7. APPLICABLE LAW. You agree You have acquired the Software Product in Hong Kong at download servers located in Hong Kong regardless of any intermediaries, companies, personnel, clients, redistribution agents, facilities of any kind or intermediate communications channels of any kind that may have been used to connect to those servers in any way, to cache, mirror or transmit files in any way or to acquire installation files for the Software Product in any way. You agree this EULA is governed by the laws of Hong Kong. In respect of any dispute which may arise hereunder, you consent to the sole and exclusive jurisdiction of the courts sitting in Hong Kong. You agree the Software Product is not licensed for acquisition, transfer or use in any jurisdiction where this EULA is not valid.

8. EXPORT AND USE RESTRICTIONS. You agree the Software Product is not licensed for acquisition, transfer or use in any jurisdiction where in the event of litigation the substantially prevailing party is not automatically granted attorneys fees and costs, payable by the substantially non-prevailing party. You agree to possess, transfer and use Software Product only in jurisdictions where the possession, transfer and use of Software Product is permitted by law. You agree to comply with all applicable international and national laws that apply to
the Software Product, including export control regulations if applicable to you, as well as end-user, end-use and
destination restrictions issued by governments that have jurisdiction over You.

9. TRIAL EDITIONS. The provisions of this paragraph shall supercede the provisions of any other paragraph of
this EULA. If this Software Product was provided by Manifold explicitly labeled in the startup dialogs as a "Trial
Edition," this EULA is limited as set forth by Manifold in its transmission of Software Product to You, or as set forth
in the startup dialogs associated with Software Product. Any such limitations shall supercede any other provision
of this EULA. Such limitations may be (by way of example and not limitation) a limited period of authorized
operation commencing from the time of installation, a limited number of invocations of Software Product, some
limitation as to the number of features enabled or processing capacity, or any combination of these or other
limitations. You may NOT modify Software Product or operate Your computer system or Software Product in a
manner that defeats the limitations set forth with the Trial Edition. You may NOT remove any dialogs or other
information associated with Software Product that set forth such limitations. Trial Editions of Software Product
may NOT be copied and may NOT be transferred to any third party. If a limitation set forth with a Trial Edition
specifies a licensed period of authorized operation ("Trial Edition Period") that is less than the 90 day period
("Warranty Period") set forth in the Limited Warranty provisions of this EULA, then the Warranty Period shall be
reduced to the length of the Trial Edition Period.

10. NO FAIL-SAFE USES AUTHORIZED. The Software Product has been created using industry standard
personal computer software technology. Personal computer software technology is not fault tolerant and thus this
Software Product is not designed, manufactured, or intended for use or resale as on-line control equipment in
hazardous environments requiring fail-safe performance, such as (by way of example and not limitation) in the
operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, direct life support
machines, or weapons systems, in which the failure of the personal computer technology employed in Software
Product could lead directly to death, personal injury, or severe physical or environmental damage.

11. LIMITED WARRANTY

- LIMITED WARRANTY. Manifold warrants that the Software Product will perform substantially in
accordance with the accompanying materials for a period of ninety (90) days from the date of receipt.
- If an implied warranty or condition is created by your state/jurisdiction and federal or state/provincial law
prohibits disclaimer of it, you also have an implied warranty or condition, BUT ONLY AS TO DEFECTS
DISCOVERED DURING THE PERIOD OF THIS LIMITED WARRANTY (NINETY DAYS). AS TO ANY
DEFECTS DISCOVERED AFTER THE NINETY (90) DAY PERIOD, THERE IS NO WARRANTY OR
CONDITION OF ANY KIND. Some states/jurisdictions do not allow limitations on how long an implied
warranty or condition lasts, so the above limitation may not apply to you.
- Any supplements or updates to the Software Product, including without limitation, any (if any) service
packs or hot fixes provided to you after the expiration of the ninety (90) day Limited Warranty period are
not covered by any warranty or condition, express, implied or statutory.
- LIMITATION ON REMEDIES; NO CONSEQUENTIAL OR OTHER DAMAGES. Your exclusive remedy
for any breach of this Limited Warranty is as set forth below. Except for any refund elected by Manifold,
YOU ARE NOT ENTITLED TO ANY DAMAGES, INCLUDING BUT NOT LIMITED TO
CONSEQUENTIAL DAMAGES, if the Software Product does not meet Manifold's Limited Warranty,
and, to the maximum extent allowed by applicable law, even if any remedy fails of its essential purpose.
The terms of Section 11 below ("Exclusion of Incidental, Consequential and Certain Other Damages")
are also incorporated into this Limited Warranty. Some states/jurisdictions do not allow the exclusion or
limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to
you. This Limited Warranty gives you specific legal rights. You may have others which vary from
state/jurisdiction to state/jurisdiction.
- YOUR EXCLUSIVE REMEDY. Manifold's and its suppliers' entire liability and your exclusive remedy
shall be, at Manifold's option from time to time exercised subject to applicable law, (a) return of the price
paid (if any) for the Software Product, or (b) repair or replacement of the Software Product, that does
not meet this Limited Warranty and that is returned to Manifold with a copy of your receipt. You will
receive the remedy elected by Manifold without charge, except that you are responsible for any
expenses you may incur (e.g. cost of shipping the Software Product to Manifold). This Limited Warranty
void if failure of the Software Product has resulted from accident, abuse, misapplication, abnormal
use or a virus. Any replacement Software Product will be warranted for the remainder of the original
warranty period or thirty (30) days, whichever is longer.

12. DISCLAIMER OF WARRANTIES. The Limited Warranty that appears above is the only express warranty
made to you and is provided in lieu of any other express warranties (if any) created by any documentation or
packaging. Except for the Limited Warranty and to the maximum extent permitted by applicable law, Manifold and
its suppliers provide the Software Product and support services (if any) AS IS AND WITH ALL FAULTS, and
hereby disclaim all other warranties and conditions, either express, implied or statutory, including, but not limited to,
any (if any) implied warranties, duties or conditions of merchantability, of fitness for a particular purpose, of
accuracy or completeness of responses, of results, of workmanlike effort, of lack of viruses, and of lack of
negligence, all with regard to the Software Product, and the provision of or failure to provide support services.
ALSO, THERE IS NO WARRANTY OR CONDITION OF TITLE, QUIET ENJOYMENT, QUIET POSSESSION,
CORRESPONDENCE TO DESCRIPTION OR NON-INFRINGEMENT WITH REGARD TO THE SOFTWARE
PRODUCT.

13. EXCLUSION OF INCIDENTAL, CONSEQUENTIAL AND CERTAIN OTHER DAMAGES. TO THE
MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL MANIFOLD OR ITS
SUPPLIERS BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT, OR CONSEQUENTIAL DAMAGES
WHATSOEVER (INCLUDING, BUT NOT LIMITED TO, DAMAGES FOR LOSS OF PROFITS OR
CONFIDENTIAL OR OTHER INFORMATION, FOR BUSINESS INTERRUPTION, FOR PERSONAL INJURY,
FOR LOSS OF PRIVACY, FOR FAILURE TO MEET ANY DUTY INCLUDING OF GOOD FAITH OR OF
REASONABLE CARE, FOR NEGLIGENCE, AND FOR ANY OTHER PECUNIARY OR OTHER LOSS WHATSOEVER ARISING OUT OF OR IN ANY WAY RELATED TO THE USE OF OR INABILITY TO USE THE SOFTWARE PRODUCT, THE PROVISION OF OR FAILURE TO PROVIDE SUPPORT SERVICES, OR OTHERWISE UNDER OR IN CONNECTION WITH ANY PROVISION OF THIS EULA, EVEN IN THE EVENT OF THE FAULT, TORT (INCLUDING NEGLIGENCE), STRICT LIABILITY, BREACH OF CONTRACT OR BREACH OF WARRANTY OF MANIFOLD OR ANY SUPPLIER, AND EVEN IF MANIFOLD OR ANY SUPPLIER HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

14. LIMITATION OF LIABILITY AND REMEDIES. Notwithstanding any damages that you might incur for any reason whatsoever (including, without limitation, all damages referenced above and all direct or general damages), the entire liability of Manifold and any of its suppliers under any provision of this EULA and your exclusive remedy for all of the foregoing (except for any remedy of repair or replacement elected by Manifold with respect to any breach of the Limited Warranty) shall be limited to the greater of the amount actually paid by you for the SOFTWARE Product or U.S.$5.00. The foregoing limitations, exclusions and disclaimers (including Sections 11, 12 and 13 above) shall apply to the maximum extent permitted by applicable law, even if any remedy fails its essential purpose.

15. ENTIRE AGREEMENT. This EULA (including any addendum or amendment to this EULA which is included with the Software Product) is the entire agreement between you and Manifold relating to the Software Product and the support services (if any) and they supersede all prior or contemporaneous oral or written communications, proposals and representations with respect to the Software Product or any other subject matter covered by this EULA. To the extent the terms of any Manifold policies or programs for support services conflict with the terms of this EULA, the terms of this EULA shall control.
Appendices

Appendices provide information on miscellaneous business, technical and corporate topics.

- **Units**: Units of measure available in Manifold.
- **Essays**: A series of essays designed to arm Manifold advocates with technological savvy with which they can win friends and influence people.
- **Help Links**: A laundry list of topics designed to catch people searching for traditional GIS notions in the index, so they can find the analogous Manifold topics.
- **Info**: Administrative, legal, sales and other information. A mix of boring, necessary and even some very interesting details, beginning with the Limitations topic.

## Tables

### Units

#### Unit Abbreviations

The following are unit abbreviations used with distance and area measurement functions in queries and scripts. In addition to these units, one can also use custom units supplied by the user as set forth in the Custom Units of Measure topic.

<table>
<thead>
<tr>
<th>Name</th>
<th>Abbreviation (length)</th>
<th>Abbreviation (area)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Angular Units</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Arc Minute</td>
<td>arcmin</td>
<td>sq arcmin</td>
</tr>
<tr>
<td>Arc Second</td>
<td>arcsec</td>
<td>sq arcsec</td>
</tr>
<tr>
<td>Degree</td>
<td>deg</td>
<td>sq deg</td>
</tr>
<tr>
<td>Radian</td>
<td>rad</td>
<td>sq rad</td>
</tr>
<tr>
<td><strong>Linear Units</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Arshin</td>
<td>arsh</td>
<td>sq arsh</td>
</tr>
<tr>
<td>Benoit Chain A</td>
<td>ch Ben A</td>
<td>sq ch Ben A</td>
</tr>
<tr>
<td>Benoit Chain B</td>
<td>ch Ben B</td>
<td>sq ch Ben B</td>
</tr>
<tr>
<td>Benoit Foot A</td>
<td>ft Ben A</td>
<td>sq ft Ben A</td>
</tr>
<tr>
<td>Benoit Foot B</td>
<td>ft Ben B</td>
<td>sq ft Ben B</td>
</tr>
<tr>
<td>Benoit Link A</td>
<td>ln Ben A</td>
<td>sq ln Ben A</td>
</tr>
<tr>
<td>Benoit Link B</td>
<td>ln Ben B</td>
<td>sq ln Ben B</td>
</tr>
<tr>
<td>Benoit Yard A</td>
<td>yd Ben A</td>
<td>sq yd Ben A</td>
</tr>
<tr>
<td>Benoit Yard B</td>
<td>yd Ben B</td>
<td>sq yd Ben B</td>
</tr>
<tr>
<td>Cable</td>
<td>ca</td>
<td>sq ca</td>
</tr>
<tr>
<td>Centimeter</td>
<td>cm</td>
<td>sq cm</td>
</tr>
<tr>
<td>Unit</td>
<td>Symbol</td>
<td>Square Unit</td>
</tr>
<tr>
<td>----------------------</td>
<td>--------</td>
<td>-----------------</td>
</tr>
<tr>
<td>Chain</td>
<td>ch</td>
<td>sq ch</td>
</tr>
<tr>
<td>Clarke's Chain</td>
<td>ch Cla</td>
<td>sq ch Cla</td>
</tr>
<tr>
<td>Clarke's Foot</td>
<td>ft Cla</td>
<td>sq ft Cla</td>
</tr>
<tr>
<td>Clarke's Link</td>
<td>ln Cla</td>
<td>sq ln Cla</td>
</tr>
<tr>
<td>Clarke's Yard</td>
<td>yd Cla</td>
<td>sq yd Cla</td>
</tr>
<tr>
<td>Decimeter</td>
<td>dm</td>
<td>sq dm</td>
</tr>
<tr>
<td>Fathom</td>
<td>f</td>
<td>sq f</td>
</tr>
<tr>
<td>Foot</td>
<td>ft</td>
<td>sq ft</td>
</tr>
<tr>
<td>German Legal Meter</td>
<td>Glm</td>
<td>sq Glm</td>
</tr>
<tr>
<td>Inch</td>
<td>in</td>
<td>sq in</td>
</tr>
<tr>
<td>Indian Foot</td>
<td>Ind ft</td>
<td>sq Ind ft</td>
</tr>
<tr>
<td>Indian Foot (1937)</td>
<td>Ind ft 37</td>
<td>sq Ind ft 37</td>
</tr>
<tr>
<td>Indian Foot (1962)</td>
<td>Ind ft 62</td>
<td>sq Ind ft 62</td>
</tr>
<tr>
<td>Indian Foot (1975)</td>
<td>Ind ft 75</td>
<td>sq Ind ft 75</td>
</tr>
<tr>
<td>Indian Yard</td>
<td>Ind yd</td>
<td>sq Ind yd</td>
</tr>
<tr>
<td>Indian Yard (1937)</td>
<td>Ind yd 37</td>
<td>sq Ind yd 37</td>
</tr>
<tr>
<td>Indian Yard (1962)</td>
<td>Ind yd 62</td>
<td>sq Ind yd 62</td>
</tr>
<tr>
<td>Indian Yard (1975)</td>
<td>Ind yd 75</td>
<td>sq Ind yd 75</td>
</tr>
<tr>
<td>Kilometer</td>
<td>km</td>
<td>sq km</td>
</tr>
<tr>
<td>Kosaya Sazhen</td>
<td>kszh</td>
<td>sq kszh</td>
</tr>
<tr>
<td>League</td>
<td>lg</td>
<td>sq lg</td>
</tr>
<tr>
<td>Link</td>
<td>ln</td>
<td>sq ln</td>
</tr>
<tr>
<td>Meter</td>
<td>m</td>
<td>sq m</td>
</tr>
<tr>
<td>Micrometer</td>
<td>mcm</td>
<td>sq mcm</td>
</tr>
<tr>
<td>Mile</td>
<td>mi</td>
<td>sq mi</td>
</tr>
<tr>
<td>Millimeter</td>
<td>mm</td>
<td>sq mm</td>
</tr>
<tr>
<td>Nautical Mile</td>
<td>nm</td>
<td>sq nm</td>
</tr>
<tr>
<td>Rod</td>
<td>r</td>
<td>sq r</td>
</tr>
<tr>
<td>Sazhen</td>
<td>szh</td>
<td>sq szh</td>
</tr>
<tr>
<td>Sea League</td>
<td>slg</td>
<td>sq slg</td>
</tr>
<tr>
<td>Sears Chain</td>
<td>ch Srs</td>
<td>sq ch Srs</td>
</tr>
<tr>
<td>Sears Foot</td>
<td>ft Srs</td>
<td>sq ft Srs</td>
</tr>
<tr>
<td>Sears Link</td>
<td>ln Srs</td>
<td>sq ln Srs</td>
</tr>
<tr>
<td>Sears Yard</td>
<td>yd Srs</td>
<td>sq yd Srs</td>
</tr>
<tr>
<td>US Survey Chain</td>
<td>ch US</td>
<td>sq ch US</td>
</tr>
<tr>
<td>US Survey Foot</td>
<td>ft US</td>
<td>sq ft US</td>
</tr>
</tbody>
</table>
Essays

These essays provide miscellaneous opinions, rants and viewpoints from the manifold.net team. Understanding these viewpoints may help friends of Manifold System in their advocacy within their organizations. In addition, some of the essays provide general information to help get more out of Manifold System.

User Interface Design
Terminology in GIS
Images can be Inefficient
GIS and Networking
What about Ajax?
Using RAM and other Machine Resources
Public Access to Public Data
Zip Codes are Not Areas
Just Say No to GIFs
Beware of Counterfeit Software

Historical Notes

A few notes of historical interest, mainly thumbnail biographies, have been scattered throughout the documentation in the following topics.

George Boole  Expressions
Christopher Columbus  Projections Tutorial
Charles-Eugene Delaunay  Transform - Triangulation
Boris Nikolaevich Delone  Transform - Triangulation
Lejeune Dirichlet  Transform - Voronoi Operators
Leonhard Euler  About Networks
Pierre Simon Laplace  Transform - Laplace
Johan Heinrich Lambert  Lambert Azimuthal Equal Area
Gerardus Mercator  Mercator
John Parr Snyder  Space Oblique Mercator
Georgi Voronoi  Transform - Voronoi Operators
User Interface Design

One of our favorite comments about Manifold System is a newsgroup posting that stated “Manifold is software for grown-ups.” A key aspect of growing up is realizing that good things require some effort. It is not at all grown up to expect that one can master a complex subject in three days. Part of growing up is realizing that study and practice are required to gain true expertise.

Manifold's user interface has been designed for the user who will be working with Manifold on a regular basis and who is willing to invest some time in mastering the system. It assumes the user will read this documentation and will learn enough about the different tools and the connections between them to choose that sequence of commands that does what the user wants.

The interface depends on the interplay of several different capabilities applied in parallel as needed by an informed user. For example, once we open a map window we will also open different panes and toolbars to do what is needed in the map. We might have a table window open and perhaps even have a drawing or image window open to work with one of the components that is a layer in the map. To accomplish various steps in our plan we click on different panes and windows and invoke different commands as needed.

This is a very different approach to user interface design than is used in “wizard” interfaces where everything proceeds step-by-step through a fixed, linear path. Wizard interfaces are fine for highly predictable, fixed tasks that use data in known arrangements. They are not at all good when a flexible approach is required.

The Manifold user interface involves use of several panes and toolbars surrounding one or more windows. It is a non-linear interface where the user sets up an activity and then chooses tools and modifies tool options during the activity. This is more difficult to learn initially but very fast and flexible for serious work.

If you have any comments or suggestions for improving the user interface, please send them to manifold.net using the process outlined in the Contacting manifold.net topic.

The Postmodern Interface

Manifold System 5.00 introduced the Manifold “postmodern” user interface, which is continued in subsequent editions. Previous editions of Manifold were similar to many ’90’s GIS systems in that they used lots of dialogs and lots of toolbar buttons. Manifold tries to minimize the number of special-purpose dialogs by relying upon global Windows ideas like Copy and Paste As and context menus to provide an upper level of command functioning. Instead of being called from lots of specific menus, dialogs involved in many operations will appear as necessary when various upper level command actions (copy, paste, context menu choices, etc.) are taken.

The postmodern interface philosophy believes it is better to use already-known, standard Windows moves in an implied way than to design explicit, special purpose procedures that must be learned. This reduces what must be learned that is unique to Manifold even as it expands possibilities. It also moves attention onto the objective of activities rather than onto memorizing the specific steps required for processing.

"Quiet Cockpit" Design

The Manifold user interface introduced "quiet cockpit" philosophy to GIS. The phrase “quiet cockpit” arises from modern human interface design for complex aircraft. The modern theory of cockpit design states that pilots should be able to focus their attention on important tasks without the distraction of numerous dials and controls flashing lights at them to announce that all is normal. In this view, a normal indication should be signaled by no signal. For example, a fuel gauge should not flash a green light to indicate there is a normal status of fuel. It should remain quiet unless fuel runs low, in which case it flashes red or some other caution color.

Manifold attempts to bring a quiet cockpit to GIS by enabling only those controls that can actually be used in the current situation. The main method is to change toolbar buttons so that only those that can be applied in the current context are visible and enabled. All other commands are hidden or disabled. Before one becomes familiar with the different tools this may seem unsettling as toolbars change when the focus is on different types of windows. After some experience one appreciates how always the right assortment of tools is at hand.

A Note for ESRI or MapInfo Users

As mentioned in the For Experienced GIS Users topic in the introduction, Manifold’s user interface is not at all like those used in older GIS products. Instead, Manifold's nomenclature and commands are based on a combination of several stylistic approaches:
Microsoft methods and terminology wherever a Manifold capability matches something within the Microsoft spectrum of products. For example, Manifold uses "Tools - Options" because that's where user preferences are kept within most Microsoft products.

- Microsoft Visual Studio, Visual C++ and Visual Basic user interface style for development and programming.
- Adobe PhotoShop concepts for photographic/artistic image editing.
- Manifold System logic and mathematics for GIS.

The manifold.net team is often asked why we did not simply clone ESRI methods and interfaces in GIS, since ESRI dominates GIS markets. There are three reasons why:

- The potential user base for GIS is far larger than the current ESRI population, which is a very small number by mass-market Microsoft standards. Making Manifold accessible to hundreds of millions of Microsoft Office users means adopting Microsoft terminology and methods, not ESRI's.
- Older GIS systems were created in a technologically less sophisticated time. Providing the full range of modern capabilities means adopting modern ways.
- We seriously intend to deliver a fusion of GIS, CAD, advanced database, image processing, photo editing, mathematics modeling and many other disciplines. The right user interface that works with a blend of such intense functions is a new user interface that blends the best ideas from all these disciplines and not just GIS.

We realize that the above approach make transitioning to Manifold more difficult for experienced GIS people who have a lot of ESRI or MapInfo experience they would like to leverage. However, for many ESRI or MapInfo users a transition to Manifold is really their first transition into the bigger world of Microsoft standards and mass-market PC software standards. For any professional person using computers this is a necessary step and not wasted effort. If you are not yet a Windows power user, we have written a Windows topic to help you power up.

See Also: Termination in GIS
Terminology in GIS

It's no secret that certain technical disciplines use pretentious words in a way that confuses newcomers. The traditional GIS community is guilty of this sin in a big-time way by employing an archaic and misleading vocabulary.

Technology in general and GIS in particular are difficult enough without making things any more difficult by using artificially obscure language. Everyone who is a friend of GIS should work hard to promote a clear, modern and easily understandable lexicon. There are over one hundred million computer users in the world working at a sophisticated enough level to be able to use GIS. Clear language (as well as mass market value and pricing) will help them use GIS.

Manifold tries to help by using Microsoft terminology to refer to points, lines and areas. This is what most of the world's mathematics and computer community uses. ESRI and ESRI imitators within GIS use different terms:

<table>
<thead>
<tr>
<th>Manifold</th>
<th>ESRI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Areas</td>
<td>Polygons</td>
</tr>
<tr>
<td>Lines</td>
<td>Arcs</td>
</tr>
<tr>
<td>Points</td>
<td>Points</td>
</tr>
</tbody>
</table>

Using the Word "Arc" is Misleading

Over one hundred million intelligent and sophisticated computer users are confused by the use of the word "arc" to mean a line. It's really misleading to use such terminology unless one is trying to discourage new users from getting into GIS. It is even mathematically illiterate.

The overwhelming usage by English-speaking, mathematically expert persons throughout the world is to use the word "arc" to mean a segment of a circle. Only a tiny handful of mathematically literate users understand "arc" to also mean a link between vertices in a graph. From a purely mathematical point of view it is foolish and pretentious to the point of being offensive to use this word in a geometric context to mean something other than a segment of a circle.

It is true that "arc" can mean a link between vertices in a graph (from which meaning ESRI apparently derived its use of this word). However, contemporary graph theory literature written in English uses the word "edge" nearly universally instead of "arc." The ESRI usage therefore is becoming ever more obscure even among professional mathematicians working in graph theory. When you hear the word "arc" in arc-speak, think "archaic".

A further effect is that much literature on "graph theory" in English now is really applied mathematics focusing on computer networks. As a result the terminology of computer networking dominates technical usage. Most people now use the computer networking word, "link." Even the word "graph" is now being replaced by the more modern word "network" when the great mass of literature is considered.

Regarding mathematical illiteracy: those relatively few people who understand graph theory are careful not to toss words like "arc" about inaccurately. It's those folks who have the weakest understanding of graph theory who leap to use this word in wrong contexts. From a mathematical perspective, only a very small percentage (likely less than one percent) of those vector objects called "arcs" within the ESRI community are really arcs. All the rest are simply lines.

A network is defined by its nodes and not by lines. The most common error of graph theory newbies is to draw a system of lines that have coincident ends and then say, "there, that's my network." Nope. If it does not have any points it is not a network. One point by itself can be a network. Two points by themselves is a disconnected network. Two points with a link between them is a connected network. Any number of lines without points is not a network even if they are arranged in a pretty diagram. Even within archaic graph theory nomenclature a line is not an arc unless it has a node at each end.

Since the vast majority of GIS data that uses line objects consists of lines drawn for purely illustrative purposes without points it is mathematically illiterate to refer to such objects in a general way as arcs. Even those few GIS data sets in which lines have points at the ends of them are not often graphs in a consistent way. Anyone who has ever had to do graph theoretic transportation analysis knows how few "clean" data sets exist in which the lines are really arcs.
Manifold avoids misusing the language of graph theory by simply referring to lines as "lines." When discussion revolves around graph theoretic concepts and operations, Manifold follows accepted modern network terminology so that the English language versions of Manifold use the word "network" instead of "graph," "nodes" instead of "vertices" and "links" to mean the connections between nodes in a network.

We realize there are many users within the ESRI community who know perfectly well what arcs are and are not in a precise mathematical way and who nonetheless choose to use this word as a matter of familiar ESRI jargon. That's OK as long as we are all careful to make it easy on newcomers by using more accepted words.

**Dump "Polygons"**

At least those things referred to as "polygons" in ESRI-speak are in fact polygons from a strict mathematical perspective. However, they are almost never regular polygons nor are recognizable as figures composed of straight-line segments. This is an important distinction to make if we are interested in advancing GIS within the general population of computer users. As regards geometric awareness, that general population can be divided into two classes: the mathematically illiterate and those who have some mathematical literacy.

Among the masses of tens or hundreds of millions of people who could potentially use GIS most have little or no awareness of mathematics and geometry. Such people have no idea what a "polygon" is and are somewhat put off by the term. However, everyone educated enough to use a GIS knows what a "region" or "area" is. Among those who have had some education in math most people understand a "polygon" to mean a regular polygon such as a pentagon or as a figure that obviously consists of a series of straight line segments. What they see in GIS contexts (curvy regions) look nothing like the "polygons" they recall from geometry class. Using the word "polygon" to describe things that do not look like polygons will unfairly confuse new users.

For this reason, a word like "region" or "area" is a better idea than using the word "polygon." Manifold uses the word *area* because this word emphasizes the inner content of such objects as opposed to the wireframe nature of regions delimited by boundary lines. We do not use the word "shape" because in the Microsoft world this word has a pre-existing meaning that is somewhat mixed between the idea of a vector object like an area and a raster region of differently-colored regions. We try to use the word *region* to refer to distinguishable subsets of pixels in images.

**Elementary Errors in Networking Terms**

There are two elementary conceptual errors within some traditional GIS packages that show up in misuse of terminology:

- Using the word *node* to refer to an internal coordinate that defines a polyline instead of a vertex in a network.
- Not realizing that a network is defined by its nodes and not by its links.

The above are dreadfully illiterate from a network math perspective and are among the errors that lead to university geography departments being treated in a patronizing way by their colleagues in the mathematics faculties. Friends of GIS should take care to use mathematical terms accurately to preserve respect for our profession. Manifold tries to help by using such terms rigorously.

Manifold can treat any system of lines and points as a network. When we discuss network subjects within Manifold, we shift terminology so that points are called *nodes* and lines are called *links*. We use the word *node* to describe points that participate in a network because universal usage within computer networking and graph theory reserves the word *node* to mean a point that acts as a vertex of a network.

People unfamiliar with networks will often refer to a system of lines such as that in the first illustration below as a "network." However, a network is defined by its *nodes*, not by its links. Links are simply a way of describing how the nodes of a network are formed into a network. If a drawing (or map) has no points, it is not a network.
Suppose we have three lines that are incident to each other at a location. This system of lines is not a network, since a network is defined by its nodes (points) and the above illustration has no points. If we would like to treat these three lines as a network we need to create a point at the ends of the lines.

We can use the Node Points transform to create a point at the end of each line. Only one point is created at the location where all three lines are incident to each other.

Note the difference between the Add Node Points and the Points transform:

Had we run the Points transform on the initial drawing we would have obtained the result above. The Points transform adds points at every coordinate location that defines the shape of the lines. The resultant points appear at what some legacy packages refer to as "nodes." At times one encounters traditional GIS software that also refers to such coordinate locations as "vertices." However that is even more illiterate than the wrong use of "nodes" since the term vertex is a very accurate, very well defined term from graph theory that very definitely means the same as node in a networking context. It very definitely does not mean the same as "inflection point" or "coordinate location in a polyline."

**Modernize the Rest as Well**

Let's face it: weird words like "coverages" mean something to someone who has been immersed in arc-stuff for the last few decades but they don't mean anything to anyone else who uses computers. There are plenty of unpretentious, good words in common usage that everyone will immediately understand that could be used instead of strange words like "coverage." Manifold uses the term drawing since that is the word almost universally used for such things by vector editors in the graphics arts and CAD worlds. Such users outnumber GIS users by about a thousand to one so it seems wise to adopt their terminology as the standard in modern computing.

In general, whenever the great big world of computing in general has a good word that is universally used Manifold tries to use it within GIS as well. At times there are no common words in use in computing but there are good English words that describe GIS things in a more accessible fashion than words used by the GIS community. If they are truly obvious words they will not confuse GIS experts while making life easier for newcomers. A few words used within Manifold are experiments that we hope will make GIS more accessible without losing any technical meaning.

An example of these is our use of the word tracing to describe vectorization. We think it is a natural word that people immediately understand that will help take the mystery out of "vectorization." It describes the process very clearly as well. Everyone understands the idea of laying a sheet of tracing paper onto a photograph and then drawing lines with a pencil to outline what is seen in the photo. The "raster" photo then becomes a "vector" line drawing.
From a purely mathematical perspective almost none of the objects in GIS drawings are really "vectors" anyway so there is no harm done in replacing this jargon with a more accessible word like "tracing". Check out the Tracing topic to see if you agree.
Images can be Inefficient

Most of the visual impressions we see published on the web and in print media are images. See the Images topic to understand the structure of images.

Images consist of a sea of pixels that are distinguished from each other only by the color of the pixel. Unlike drawings, there are no "objects" in images. Any objects we see in images are really illusions in the human mind. When we see a cloud of pixels that appears to be a line from far away, the appearance of the line is an interpolation created within the human eye and brain physiology… there's no "line" actually there.

Newcomers to GIS are often seduced into using images inappropriately for a variety of reasons:

- Images are easy to create by scanning photographs or paper maps.
- When scanned from paper maps, images can have a very rich visual appearance.
- Creating new drawings with high detail is a tedious and time-consuming process.
- Formatting drawings to provide a rich visual appearance is very time consuming and requires good taste and expertise.

However, one can pay a high price for using images inappropriately within GIS. There are many circumstances where using images is the right thing to do and there are times when using images is a very bad idea. One must be aware of the benefits and costs of using images. In particular, using images as a replacement for vector drawings is rarely a good idea. Using images for photographs, scanned data samples and other continuously varying, analog data representations is fine.

Since the usual problem is inappropriate use of images, the following sections focus on the downside of using images.

Large Size / Low Information Content

When used to replace vector drawings, images can be unacceptably large for the amount of information they contain. Let's see how by doing a thought experiment. Suppose we have a room with a large, smooth floor like a basketball court and we want to draw a line on the floor that shows an outline of Europe. We could proceed in at least two ways.

One way to draw the line would be to place thumbtacks at places marking the outline and to then stretch a ribbon or string between the thumbtacks to mark the outline in a "connect the dots" fashion. This method is analogous to how a drawing works in that the only data necessary to keep on hand are the coordinates that mark the shape of the outline.

Another way would be to cover the entire surface of the floor with small, flat pebbles. We would use white pebbles throughout the entire floor except for those places where we wanted to mark the outline of Europe. To mark the outline, we could carefully replace some white pebbles with black pebbles. If we placed the black pebbles carefully, we could view the floor from an elevated location and see the outline of Europe in the pattern of black pebbles winding its way through a sea of white pebbles. This method is analogous to how an image works. The pebbles are the same as pixels.

It takes no special genius in computer software design to consider the above situation and realize that marking a line on the floor using a ribbon to connect the essential locations that define the shape of Europe is a lot faster and more efficient than placing millions of pebbles on the floor. It is also less wasteful of resources. In the case of using pebbles, most of the "pixels" placed on the floor are not necessary. They simply provide a sea of white against which the black pebbles become visible to our eye as a pattern that we see as the shape of Europe.

Note also that in the first case we have a true object, the ribbon, which makes up the outline of Europe. In the second case there is no object, just a visual effect we reckon to be an outline by virtue of contrasting color.

Suppose we had a computer file that told us where to place thumbtacks in the first method and we also had a computer file that listed all the locations and color for each pebble in the second method. The file for the first method would have to list only the number of coordinate locations needed to define the shape of Europe, perhaps as few as a few thousand numbers. With the pebbles, though, whether or not the pebble participates in showing the actual shape of Europe, each pebble location and color must be noted. The computer file for the second method would therefore use millions of numbers to contain the same information as the "drawing" file.
The situation in GIS is very similar. A drawing showing urban streets in high detail might be only five megabytes in size. An image showing the same streets as a raster could easily be 125 megabytes in size. Although images can be compressed to eliminate some of their waste, it seems unreasonable to first choose a highly inefficient method and then spend a lot of time trying to make it somewhat more efficient. Compression also does not eliminate the need to manage pixels in uncompressed form when the images are actually used within a GIS.

There are cases, of course, where images can be an efficient way of presenting data in terms of the amount of information captured for a given size of file. The obvious case is photographic images. Suppose the floor of our hypothetical basketball court was covered with an enlarged photographic image of a region shot from space. Each "pebble" or pixel could be the right color to form an overall photographic image. To capture the same image using a drawing would not gain any efficiency since the drawing would have to contain at least as many points as pixels in the image.

In another example, suppose we wanted to present a detailed pattern of data such as minute variations in temperature or reflectance over our basketball floor. In that case, a vector drawing representation would have to put points at every location we wished to note together with a measurement. There would be as many vector points as there would be "pebbles." In this case as well an image makes sense. In general, where data represented is analog or continuously varying in nature images are a good choice.

**Low Accuracy**

To consider our example of two ways of drawing an outline of Europe, it's clear that using the drawing approach provides precise accuracy for the position of every thumbtack marking the outline and the ribbon or string that passes between it. In contrast, when marking the outline of Europe with black pebbles placed within a sea of white pebbles we will inevitably confront "the jaggies" where the pattern of black pebbles stair-steps between pebble locations.

No matter how far we zoom into a vector drawing the accuracy of the shapes drawn between coordinates is perfect. A line drawn between various coordinates will always be perfectly razor-sharp at any zoom level.

With images, in contrast, we can always zoom into the image to the point that pixels appear as large square shapes. At that point, it is not exactly clear where lines may be located. We may think we can interpolate by eye but that will not work if we zoom even further into the image. Images are therefore "accurate" only if they are seen from far enough away that our eye doesn't see the inaccuracy of the image.

A further point of inaccuracy is that most images representing vector maps are derived from paper maps, which in turn were created by human, subjective interpolation of vector data. For example, terrain elevations are usually measured at specific points. When those points are marked out and joined by contours, the contour drawing is inevitably a process of interpolation. When maps are printed, they are further interpolated into the pattern of small dots used in the printing process.

To convert a printed map into a digital image, the printed map is interpolated yet again when it is scanned. The result is a highly irregular pattern of pixels like that seen above (taken from a USGS digital image of a scanned paper map) that represents several levels of interpolation from whatever original data was used to create the map.

**Limited Data Content**

Drawings can contain arbitrarily rich data for each object. In a drawing, each object is connected to a record in a database table. That table can contain many fields of very rich data types, and the table can be linked to other tables via relations. Drawings can therefore act like visual windows into a database. The database can also act as an algorithmic window into the drawing since objects in drawings can be selected using SQL and other database methods.

Images, in contrast, can contain no data other than the color of their pixels. With an image there is no connection to database information and little opportunity to select except by color.
Drawings also have the advantage that the specific coordinates that define the objects within the drawing also define precise spatial relationships between those objects. The drawing provides a rich set of implied data by virtue of arrangement of the objects it contains. One can use drawings to compute relationships such as how much of a particular object is contained within a different object, or to find the longest line or the largest area.

Images have no such rich spatial relationships because they have no objects, only pixels. There are no objects in the image so there is no way to say where one object ends and the next begins. With images, where an object begins or ends or whether it exists at all is a matter of opinion, not geometry.

Some enthusiasts of an exclusively raster approach to GIS may complain that there are methods for assigning database information to regions of pixels and to “classify” or otherwise assign object characteristics to regions of pixels. However, what is really going on in such cases is the creation of a set of meta-information that is really a type of vector drawing. It is more efficient to simply use real drawings, perhaps in combination with images.

Restricted User Lifestyle

When using images to represent vector data better represented as drawings one often ends up with file sizes that are larger by a factor of twenty or more than the equivalent vector data set. Manipulating such large amounts of data takes profoundly more computer resources than is required for the vector drawing equivalent. The result from a user interface perspective is that operations with large images will be so slow that interactive work will be difficult.

When operations run many times slower the effect is a reduction in the quality of life of a user. A fast, instantaneously responsive user interface is a joy to use. It enhances the mental engagement of users with their work. A slow, tedious interface that requires pauses of many seconds between all operations causes unhappiness and stress. Using images inappropriately as replacements for vector drawings is a pathway to unnecessary stress.

Financial effects for using images are a real effect as well. When every layer in a map requires over one hundred megabytes the machine required for operation will be considerably more expensive that that required when every layer is a mere five megabytes in size. Because the number of pixels to be processed in an image goes up by the square of any increase in linear dimension, the processor speed required increases much more rapidly than the apparent increase in size of an image. Images that appear slightly larger could require twice the processor speed to maintain the same level of system responsiveness.

Therefore, the costs of running large images go up faster than the size of the images that can be reasonably used at various machine price points. Large images also require large amounts of RAM and larger hard disks for storage. Finally, projects involving large images are more difficult to transmit over Internet or private wide area networks for collaborative work than projects based on equivalent drawings.

Advice to the User

- Avoid using images to replace vector drawings. Instead, make a special effort to acquire the necessary vector drawings and invest time into formatting those drawings.
- If no drawing exists and only an image is available to represent a vector drawing (such as a scanned image of a paper map), invest the resources required for creating a drawing from the image. Use Manifold's tracing tools or hire a digitization contractor to create a drawing version.
- Sometimes, of course, it makes sense to bow to expediency and simply use a fast machine and a scanned image of a paper map.
- Use images for photographic data.
- Use images for data sets that represent continuously varying data such as terrain elevations, temperatures or other intrinsically “analog” data.
- Use images in combination with drawing layers to enhance presentations through artistic effects.
GIS and Networking

"A little learning is a dang'rous thing." - Alexander Pope

Few modern phenomena have taken hold as rapidly as the recent fashion for networking technologists to make money by taking advantage of the greed and stupidity of stock market investors. There’s no better way to start a stock market craze than by leveraging an exotic new technology or business method that the average investor does not understand. Add some pseudo-scientific gloss and you can take them to the cleaners every day. Even after a dot-com bust there are plenty of investors lined up to pursue the next big new thing.

The opening years of the new Millennium are an especially fine time to fleece investors with new Internet schemes or network-related glitz. At the time of this writing much of the world is riding a wave of immense prosperity compared to previous decades. Even with stock market declines, lots of people still have lots of money to invest. At the same time, math and science education in the US and many developed countries is at an all-time low. Never before in history have so many poorly-educated people had so much money they are willing to hand over to confidence men who can dazzle them with big words and mighty promises.

"GIS over Internet" is the GIS industry’s own contribution to fleecing the ignorant. By this phrase we don’t mean publication of images but rather the idea that GIS in local machines will be replaced by GIS done mainly on applications servers working with dumb, local clients.

As an Internet company that sells Internet Map Server software we would be the last to suggest that Internet is not in fact a revolutionary new way of communications. To compare Internet to the invention of movable type and its popularization by Gutenberg may actually be an understatement. Internet and the technologies it spawns will in our view result in a greater transformation of human society than the appearance of fast and easy printing technology.

However, just as the simple act of printing does not guarantee infinite wealth, just so simply using Internet does not guarantee sustainable wealth. To continue to make money in the long run one must use Internet sensibly. So as not to accidentally invest in really dumb ideas that are being promoted to take advantage of the Internet craze we have to understand some of the technology involved. Investing based on technical reality will help us keep our eyes open during market run-ups drive by speculative idiocy and will also serve us well in avoiding losses when investments based on fantasy collapse. Even if investing in stocks is not the game, understanding the fundamental conceptual errors behind GIS over Internet schemes will help organizations avoid the expensive mistake of investing in such schemes for their own operations.

Processor Speed and Internet Speed

Local area networks have continued to increase in speed to keep pace with higher speed local disk accesses. Small businesses and homes have shifted to hundred-megabit speeds and are beginning the move to gigabit networks while large businesses have already shifted to gigabit networks. People are using the speed of these local networks to make it much easier to reach out and get data regardless of where it may be stored. Modern local networks are therefore great at creating a transparent group computing experience when it comes to disk accesses.

However, disk accesses are only part of the picture of what makes for a modern computing experience. As fast as modern disks are, they are a thousand times slower than the fast parts of the computer. Hard disks typically work a few thousand times slower than dynamic RAM. RAM in turn works many times slower than the processor driving the system. Comparing the speed with which a processor talks to local RAM and local video memory to the speed with which a hard disk operates is literally like the speed difference between a fast bullet and a slow human walk. See the Using RAM and other Machine Resources topic for what this means to an application user.

This speed difference is an important factor because the human perception of computer response (whether we judge an application to be fast and responsive or slow and unusable) is largely driven by what happens at processor, RAM and video subsystem speeds. The practical limit on how cool or intelligent or responsive a modern user interface may be is largely determined by the speed of the high speed parts of the computer. This is something every 14 year old knows who has stepped up from playing a game on a slow video card on a slow processor to rocketing along with a fast processor and a big-time graphics engine with lots of megas of video RAM.

The speed of the hard disk or network has zero influence on the quality of the immediate user experience, because both are far too slow to participate in the user experience even in the slow case.

This speed differential calls up another factor that many GIS over Internet and Network PC investors fail to understand: As slow as hard disks and local networks may be compared to the bandwidth between processors and local RAM, Internet is many, many times slower. Consider that all the exciting talk in recent years about "broadband" Internet connections boils down to pathetically slow speeds, on the order of a megabit per second or so, a thousand times slower than a fast local network and usually millions of times slower than local system internal function.
If the difference between processor - RAM speed and hard disk speed is more than the difference between a speeding bullet and a human walk, then the difference between hard disk speed and Internet speed is literally like the difference between a human walk and the slowest crawl of a snail. Even a very fast connection to Internet is literally a thousand times slower than a local hard disk, and it is approximately a million times slower than local processor - RAM accesses.

Forgetting this key technical difference leads to business mistakes that are as dumb in their own way as the decision of the Tennessee state legislature one year to vote to make "Pi" a round 3.00... it just doesn't work. The plain facts of the matter are that Internet speeds are a million times slower than speeds achieved between a processor and its local memory. More importantly, although the speed at which processors and local memory communicate are getting faster every year Internet speeds are growing in speed at a much slower rate of improvement. This means that the arguments presented in this essay become stronger every year, not weaker. Quite soon, as processor - RAM speeds continue to rapidly increase while Internet speeds hardly increase at all, the difference between "in the box" speed and Internet speed will be like the difference between a fast bullet and a fast-growing plant.

**The Uses of Speed**

The main reason one needs speed is to support a high quality, modern user interface. A fast, elegant user interface tosses billions of bytes back and forth between processor and RAM to make the user experience smart and easy. Manifold System, for example, runs thousands of functions as a mouse drifts across a map. Every one of those functions depends on visual positioning and other factors to make the user interface more intelligent. Because human time is much more expensive than processor cycles it makes sense to throw processor cycles by the billions at even small tasks just to save a little bit of time for humans. The computer should have to work very hard so humans need not work so hard or study as much.

Even the fastest Internet connection is thousands of times too slow to support such sophisticated user interfaces. Remember that a processor - RAM connection is like a bullet to a walk when compared to a hard disk, and the hard disk is like a walk to a slow snail compared to Internet. Folks who make the technical mistake of thinking that Internet can support the same user interfaces as a processor - RAM connection are making as dumb a suggestion as it would be to suggest that a snail could play on a World Cup soccer team. Actually, it's worse than that. They are suggesting a snail can keep pace with a fast rifle bullet.

This is the main reason that GIS over Internet cannot handle the modern interfaces expected of modern applications. Internet is simply too slow. This is also the reason people do not do Adobe PhotoShop or other graphics editing across Internet, nor do they do even low-bandwidth tasks like word processing over Internet. Internet is just too slow. This is something everyone knows who has any actual experience of "Internet applications" such as web-hosted email. Such applications feature frightfully slow and stupid user interfaces that are hampered by the very slow speed of Internet. [They are also hampered by the "tragedy of the commons" effect whereby Internet connectivity is always automatically overloaded, but that's a different essay.]

GIS across Internet proponents often respond to this point by making a really stupid comment. They point out that Internet speeds are increasing every day. That's dumb because the comparison is not between Internet speeds of today and the speeds that Internet might achieve tomorrow. The comparison is between Internet speeds on any given day and the speeds achieved between processors and RAM "within the box" on that same day. Internet is getting faster, but processors and "in the box" speeds are getting faster still. The comparison is not to a fixed target, it is to a moving target driven by relentless speed increases in the link between processors, RAM and video memory.

Sure, modern DSL or cable modern Internet connections are fast enough to support user interfaces that are better than those on time-shared, Teletype printers or plain vanilla ASCII terminals as were common in the early 80's. So what? People are no longer satisfied with such interfaces. The requirement is not to grow Internet speed so that one day it will equal the obsolete user interfaces of the past. The requirement is to equal the elite user interfaces of modern times. It is a race in which Internet speeds are falling farther behind.

This is the key technical disconnect that wishful thinking and glib marketing often hide from the unwary. As technologists, we fully expect Internet and Java applets and such to get a hundred times faster in the next five years. Big deal. That's still a million times slower than the bandwidth achievable today by using local processors and local memory, and the Internet speeds of five years from now will still be thousands of times too slow to do a really first rate user interface by today's standards, let alone by tomorrow's standards. If one expects the standard for "power" user interfaces to improve in the future, then future Internet speeds will be even less suited to serve the user interface standards of tomorrow.

One main reason that Internet speeds are not increasing as fast as processor and memory speeds are increasing is that the speed of processors and RAM is driven mostly by local advances in silicon lithography and fabrication that are unconstrained by large scale infrastructure. Any improvement in any silicon fab quite rapidly circles the
The Network PC is simply the multi-user idea of a central processor and dumb terminals brought back from the
broader audience if the individual vellum sheets were razored out of the book, mounted in a pretty mount and sold
could be sold for a few thousand dollars as a whole book could be sold for tens of thousands of dollars to a much
completed data set. About 30 years ago rare book dealers discovered that a hand-written book on vellum that
The money factor in GIS over Internet has to do with the magic of selling the same thing piecemeal instead of as
the UNIX dream of a universal, quasi-dumb X terminal attached to a server into a modern
determinants of all technology: politics and money.

A final restriction on the speed of Internet for interactive computing processes is the speed of light. Already,
computers are so fast that speed-of-light issues are beginning to arise even over the short distances within a
computer. The physical distances between a client and a distant Internet server impose a crushing limitation on
speed as a result of speed of light travel time between the two, over distances that are millions of times farther
than intrachip distances. Although this may seem to be science fiction to the technically illiterate, something that
might be erased with a suitably glib disclaimer in marketing documents, the speed of light limitation is a genuine
factor in high technology that cannot be wished away.

In all fairness, Internet is fine for “batch” operations that are not interactive. It's also fine for presentations and
light-duty user interfaces. However, the issue is not the complexity of “batch” spatial operations that can just as
easily be done on a server as on a local workstation. The issue is the quality of a sophisticated, interactive user
interface, which is far more bandwidth intensive. One can easily write a fearsomely complex query in a few text
lines of spatial SQL. The point of modern GIS is avoiding the use of dumb text interfaces for such things
whenever possible and instead substituting a visual, seamless, point-and-click approach. Again, the point is to
use infinite machine cycles to save us humans from having to do drudge work.

We should also note that giving up one's GIS solution to an off-site server is a sucker trap that defies human
nature. It is simply asking too much of human nature to believe that any server operator will not over-sell and
under-power the server. As anyone who has ever used Internet knows, they don't call it the “World Wide Wait” for
nothing. In modern times a multi-gigahertz processor is so cheap that one now buys such supercomputer-class
machines for one's children to use on schoolwork. Who would want to timeshare any machine when we can have
our own personal supercomputer for lightning response whenever we want it? The buzzword for this in
organizations is "scalability" - if each new user brings a supercomputer class machine to the party, we can keep
adding users as long as we want without performance problems.

We think Internet is great for selling product, for delivering data sets, for synchronizing work groups, for publishing
summaries of work done and delivering updates and all sorts of work that does not require an intensive,
interactive user interface. Products like Manifold's Internet Map Server capabilities are great for such things.
However, if one wants a smooth, intelligent, sophisticated, fast, and assisted user experience, it's clear that
Internet-based applications will lag far behind modern technology applied "in the box."

Willing Slaves

So, why does one hear so much about GIS over Internet? We think it boils down to those two eternal
determinants of all technology: politics and money.

The politics are partly about Bill-envy and a desire to outflank Microsoft. The ill fated push into Network PCs
(billions of dollars down the drain on a really dumb idea) by an alliance of Microsoft competitors was motivated
mostly by a desire to wedge a crowbar between Microsoft and users. The idea of a Network PC is that all
intelligence and power should reside in a central server, which serves web pages to low-power machines that do
nothing but operate network clients.

The Network PC is simply the multi-user idea of a central processor and dumb terminals brought back from the
dead. It is recasting the UNIX dream of a universal, quasi-dumb X terminal attached to a server into a modern
lexicon of "Network PC" and "Application Server." The product-marketing basis for pushing Network PCs was
that it is easier to pry users away from Microsoft if the only application that needs to be ported is a browser. The
Network PC failed, of course, because neither PCs nor their users are stupid. The benefits of becoming a willing
slave to a central organization are far outweighed by the benefits of distributed, independent use of one's own
intelligent machines.

Another set of politics pushing for "Internet applications" revolves around big IT organizations discovering a fine
new tool to centralize everything once more. It may or may not be convenient to have someone else do system
administration but it is seriously convenient for organizational empire building when all computing activity is
centralized within the organization that runs the servers.

The money factor in GIS over Internet has to do with the magic of selling the same thing piecemeal instead of as
a complete data set. About 30 years ago rare book dealers discovered that a hand-written book on vellum that
could be sold for a few thousand dollars as a whole book could be sold for tens of thousands of dollars to a much
broader audience if the individual vellum sheets were razored out of the book, mounted in a pretty mount and sold
off one by one to collectors. Within a few years, an entire generation of manuscript books disappeared onto the walls of middle-class collectors as framed, single page ornaments.

Serving single images of maps over the web earns more money for database owners (either in a pay per view or as a lure for advertising) than selling the maps as complete digital databases. It's also a great way for the owners of digital map databases to retain ownership of the real GIS content. For public agencies who have come to enjoy selling public data for high prices it's a great way of being able to say they are meeting their mission of making public data available while in fact denying the public access to the real data. This allows the agency to charge high prices for data on CDs. It also allows bureaucratic expansion of programs and personnel needed to "develop" web pages instead of the relatively simple task of publishing data sets.

Another important factor in the money side of the equation is that there's a lot of financial pressure on traditional GIS companies to transform themselves into something other than a "old economy" software vendor. It can't be easy for someone who has spent decades building a GIS company to see a group of twenty-something web entrepreneurs achieve a valuation of a billion plus dollars in less than a year, merely because they thought up a better Ponzi scheme involving a dot com, or an LBS play or wireless play or whatever. Even with stock market declines, companies that can put a network spin onto their business are valued more in today's stock market than are ordinary software companies. Selling oneself as a "Maps over Internet" player is a great way to increase valuation.

Selling high and mighty web service ideas to technically illiterate investors is also easier work than competing with aggressive fanatics in software markets. If you were running a traditional GIS company, what would you prefer: fleecing Wall Street with web / wireless talk or risking financial ruin by getting into a price war with the likes of manifold.net? No wonder some GIS companies have promoted "GIS over Internet" like their lives depended on it.

We would like to point out that some institutional users with the best of intentions trick themselves into thinking GIS over Internet is a cost effective idea based on the notion that a browser costs nothing while a real GIS package can cost a lot. If the cost comparison is against some ancient ESRI software item that costs thousands of dollars a seat we can understand how such a cost comparison might be persuasive. However, if the cost comparison is against something like Manifold, which is inexpensive to begin with and heavily discounted in volume, then the idea of constructing an expensive web-interfaced GIS is a false economy compared to deploying Manifold on client desktops.

The web-based GIS will have a poorer interface than Manifold, so the human cost of actual use will be higher (immediately eliminating any savings of using free browsers compared to Manifold license fees). A more directly visible cost will be the cost of developing, deploying and maintaining a semi-custom web-based GIS system. This cost will greatly exceed the cost of Manifold licenses. Worse still, the Manifold licenses are a one-time cost while the maintenance and continued development (after all, we do want to keep up with progress in software) of the web-based GIS will be a massive recurring cost in every annual budget.

The Manifold Way

At Manifold we believe that a cool user interface written by experts that takes maximum advantage of "in the box" power will always far outperform a GIS over Internet solution written by equally smart experts. For serious GIS work the best user interfaces and the fastest, largest and most sophisticated spatial work will happen within your local workstation.

Web interfaces are fine for casual use to serve up "pay per views" and other visual summaries of GIS work done in the box. Internet is fine for such things and is a good way to provide reports and pictures of one's work to a worldwide audience. It's also a great way to publish a project that allows simple user interfaces for browsing and elementary queries. That's why we have an Internet map server within Manifold: for such uses it's a great idea.

For real work with real data, even simple work with simple data, corporate users will always have the best experience using a GIS like Manifold locally even if the data is remotely archived. Architectures like Manifold System Enterprise Edition allow users to gain the archival and control benefits of centralized data warehousing while retaining the performance and local response benefits of truly distributed computing, where the application runs locally "in the box."

It is also important that GIS consumers use their experience to argue against schemes by public agencies that bottle up public GIS data behind a "webstacle" that merely serves images. Don't stand by while someone says spoon-feeding a few images is adequate public access to public data. Point out that if the GIS data itself is not made available, giving a summary report in the form of a web-served, dead image is not an acceptable substitute.

There is no corporate IT argument for GIS over Internet that does not apply with equal force to other corporate asset bases and activities such as graphics arts, images, documents, spread sheets and other corporate data.
Whenever you encounter a “GIS over Internet” proponent, take a moment to ask them why GIS over Internet is such a good idea when PhotoShop over Internet or Microsoft Word over Internet or Excel over Internet is such a lousy idea.

Above all, remember the primary issue is not whether GIS over Internet is a good or bad thing. The key issue is whether it is worse or better than expertly implemented alternatives. It is also important to think clearly and not be fooled by whatever is the latest innovation in fleecing investors. See, for example, the What about Ajax? topic.

See Also

What about Ajax?
Using RAM and other Machine Resources
What about Ajax?

Some readers may come away from the previous GIS and Networking essay thinking that it no longer applies now that Ajax is in the world. That reveals a deep misunderstanding both of the previous essay as well as of Ajax, hence this essay to clear the air.

The short form of this topic would simply state that we are happy the web community agrees with the conclusions of the previous GIS and Networking essay and by embracing Ajax has finally and so whole-heartedly adopted the recommendations that essay first set forth several years ago.

But that would be no fun, as some Ajax advocates seeking the end of local applications do not really know what "Ajax" is and thus think it opposes or is an alternative to the notions in the previous essay. Their minds are often so confused by the acronym they don't realize they are in violent agreement with developers of local applications such as manifold.net. Precisely because manifold.net is indeed supportive of the real idea behind Ajax we would like to make it clear at the outset of this essay that the Ajax idea is our mantra, an idea we have put into action for many years.

Hotheads who don't bother reading the previous essay, this essay or who misunderstand their contents should therefore refrain from sending pointless flames to manifold.net accusing the company of not supporting such a wonderful, politically correct thing as Ajax. Such hotheads should avoid spontaneously combusting over matters that are not in dispute.

As a corollary, it seems that people's misunderstanding of this essay as well as their willingness to lose money in web application schemes using deceptive sound bites about Ajax is directly proportional to their failure to grasp the technical concepts set forth in the GIS and Networking essay. Therefore, that topic should be read first as it is the foundation for this essay about Ajax.

What about Ajax?

One hallmark of web-based applications promoters is their never ending ability to come up with new acronyms to confuse weak minded investors. "Ajax" is the latest acronym to enter the fray, and serves mainly to prevent imbeciles from noticing that a company has abandoned the idea of web-based applications (because they are too slow and barren) and has instead decided to get into the business of providing software that runs locally (because it is fast and rich). Like all such good deceptions, the acronym is wrapped within a cover story that evades precise definition and employs politically correct words that rubes can use to impress their friends without knowing quite what they mean.

(We assume the reader has read the previous essay and therefore knows with technical precision exactly why web applications run too slow and are barren in comparison to local applications that run fast and are rich.)

Ajax stands for "Asynchronous JavaScript and XML," referring to the putative theory of the acronym. The story begins by noting that on the one hand, people don't like long pauses in web applications while big images required for GUI maintenance or other information transfers occur, and on the other hand people are not usually continuously active when working with a web application. For example, someone might request a page of text from a web site and then spend time reading it and so not really be commanding the application to do anything while he or she is passively reading.

The Ajax story tells investors that the way web developers will get around the seemingly insurmountable technical problems listed in the GIS and Networking essay is to take advantage of the "dead times" when people aren't actively working. The idea is that the time-consuming action that makes web applications so lethally slow can be accomplished during the dead time when people aren't actively working, and so the user won't notice the application is lethally slow. This, of course, is a fairy tale. But like all good fairy tales it leverages some kernels of enduring truth.

The key fallacy is failing to realize that significant interactive applications such as PhotoShop don't have enough dead time to make up for the relative disadvantage of web applications compared to local applications, so just saying that we will toss images in advance of need doesn't quite do the trick. Ajax proponents concede that point, and therefore take their story further into territory that is not at all a fairy tale. The heart of the matter with Ajax is very true and something that Manifold very much puts into practice.

The Ajax story continues to say that the way to get around web limitations for web applications is to download the application into the local computer, where it can execute with the blazing speed and rich capabilities made possible through the use of distributed processing, local processor-to-RAM bandwidth and so on. You won't usually see it described that way, because Ajax partisans don't like to acknowledge the concepts set forth in the GIS and Networking essay as their reason for being. Instead, they say things such as "JavaScript running locally can be updated from time to time to avoid loading the network." But what they are talking about is not attempting
to run an application through the web at all but instead downloading it into the computer where it can run as a local application. The essence of Ajax is that local applications are better than web applications when rich applications are required.

The matter cannot be stated that clearly by web-centric companies because if it were, the web company promoting Ajax would see a decline in stock price as trend-conscious investors realized that the company was abandoning its essence as a web company and surrendering to Microsoft. No web company worth its inflated stock price will openly admit that Microsoft was right all along and that rich applications for the most part should indeed be run locally. So a convenient acronym is pressed into service that technically illiterate investors won't understand means "I'm doing the opposite of what I say." Using misleading acronyms is a time-honored way for Silicon Valley insiders and Wall Street professionals to fleece both the credulous masses as well as lethargic institutional investors.

The third part of the Ajax story line is the use of XML as the communications method between the local application and information tossed to it by some central web host. This and the explicit citation of JavaScript is nothing more than pandering to the insularity and stupidity of anti-Microsoft malcontents (there's one in every crowd) who would like to see the idea of democratic, distributed applications under every person's own control replaced by a centralized web application controlled by an elite web bureaucracy. Such people hate Microsoft so much they are willing to do whatever stupid thing comes along that promises to lever power away from Microsoft and from the desktop applications that Microsoft utterly dominates. Such self-destructive behavior, of course, assures that anti-Microsoft malcontents remain a tiny minority, a fringe element hovering around 5% or 6% of the worldwide base of computer users.

This fringe element does enjoy visibility despite their small numbers because for competitive reasons they are aided and abetted by companies who compete with Microsoft, such as SUN, who are forever hopefully coming up with new tricks and baubles, like Java, that don't really change anything but which they hope will provide a rallying cry for their partisans and a distraction for Microsoft. A deceptive way of presenting Ajax is one such trick.

Real men, of course, compete on the merits and don't resort to evanescent tricks. As Tacitus says, they "fight sword in hand, taking their wounds in front." In computer terms that means not saying you are doing a web application when you are really delivering a local application, and it means doing a better job in local applications, which continue to be the decisive battlefield, than your competition. That takes real effort and if you lose that contest no amount of smoke and mirrors will prevent an ultimate decline in market share.

For example, the result of the fundamental combat between SUN and Microsoft in the main contest over local applications is that SUN took their operating system and their applications ecosystem in hand against Microsoft's operating system and applications ecosystem and SUN decisively lost. SUN overwhelmingly lost despite enjoying a starting position much more powerful than Microsoft's when combat was first joined. Almost everything which followed that decisive loss is just a sideshow, as continues to be demonstrated by the plummeting fortunes of SUN.

As a side note, it is interesting that for all of its flaws Apple stands alone as one of the few system alternatives to Wintel, partially because Apple has continued to manfully compete with Microsoft through head-on competition over the merits of desktop applications, sword in hand wielding an applications suite of exquisite elegance and style.

To return to Ajax and the choice of language and communications protocol embedded in the acronym, experienced computer science professionals know that the choice of a particular language or a communications protocol is irrelevant except to the degree that it affects measurable parameters such as performance. If anything, citing a particular language or protocol reveals an amateur fascination for transient trivialities over paradigm-shifting substance.

On the trivialities front, an interpreted language such as JavaScript is a strikingly poor choice for efficient code exchange and an appallingly bloated format such as XML may be the worst choice for a machine-to-machine communications format imaginable, second perhaps only to Braille, in terms of raw efficiency. There's no point in bloating up a communications medium to make it "human readable" when it will be used only for one application to exchange data with another application.

But then Ajax as it is misrepresented is not really about computer science, it is about politics and fleecing investors, and so one must use technologies that serve those purposes rather than efficiency. JavaScript and XML are perfect for that purpose as they deliver a guaranteed anti-Microsoft pedigree and by their very sound convey the required attitude of disdain for desktop applications.

Since few people in web applications really know what Ajax means and since many people saying they use Ajax really mean only that their web application is written in JavaScript and uses XML in some moronic way, let's take a look at a true Ajax application, Google Earth, to see how the Ajax thing is done by bona fide experts.
Google Earth is the newly famous “web application” that allows people to fly about visual representations of the Earth with amazing speed. The application allows people to move from place to place, using a highly appealing GUI that seems to let users zoom from one region to another as if they were flying in from space to see details of a particular place. Google Earth’s claim to fame among investors and web application proponents rests upon the belief that the technical genius of Google, using Ajax, has made it possible for a web application to be as fast as a local application. That would be an amazing story if it were true.

The truth, of course, is that Google Earth is fast not because it is a web application but rather because it is a local application, an enormously large one, that is downloaded into the user's computer when the user signs up for Google Earth. Thereafter, the application is updated from time to time just as any local application might be through updates or service packs. Most functions of Google Earth are executed locally “within the box” for the reasons discussed in the GIS and Networking essay.

It’s true that Google Earth expertly employs the Ajax GUI shell game of moving imagery into position while the user's eye is occupied doing something else so that the user does not notice the lag. Just as a seasoned magician distracts the audience's eye with one move while covertly building an illusion with an unnoticed hand, such shell games are a part of every expert programmer's repertoire of GUI techniques. This is one of the key ideas of Ajax, but when people describe it as something new one wonders just how negligent their educations must have been for them not to know this sort of thing has been a routine part of programming for many decades. Even so, it has rarely been done as deftly as by Google Earth.

In Google's case, that is why when a user wants to jump from viewing one particular region to another Google Earth insists on zooming out from that region to a viewpoint in space and then zooms back down from space to the new region. The time spent zooming out to space and then back down to the new region keeps the user occupied with a pleasant joyride into space while Google Earth begins downloading part of the new, detailed image that is required. On our way out to space and back we don’t notice how much time has elapsed.

When the user zooms back to Earth the more detailed image already has some central tiles streamed into the application, and while the user's eye is drawn to the center more peripheral tiles and more details of the image can be streamed in, hopefully with less notice of a time lag than if the entire image were brought in at once. All this works very well and is impressive, a truly masterful GUI that combines the psychology of user perception with the power of local applications and the utility of fetching small data from the web.

But the key notion is that Google Earth is indeed a local application of the sort that Microsoft has built for years. The speed and power of Google Earth arise mainly from the idea of distributing applications away from centralized, time-shared servers into local desktop machines. It is, of course, a local application that is downloaded via the web, but that's something local applications have done for many years, a valid use of the web as a distribution medium. Likewise, it's true that the application fetches data via the web to display, but again, that has been done for many years by local applications. As we’ve argued in the GIS and Networking essay the web is a fine medium for distributing applications or for delivery of relatively low-bandwidth information.

The core idea remains that Google Earth, like all functioning “Ajax” applications, is really a local application running on the local machine, not a web application running on a central server. Running rich applications locally is the core idea in the GIS and Networking essay and is the core idea behind the personal computer revolution that killed off centralized, time-shared minicomputers. It is still the core idea that keeps companies like Adobe and Microsoft and AutoCAD thriving. And now it is Google's turn to discover that rich applications can be done faster and better by harnessing through local applications the distributed power of billions of desktop processors. The question is how can Google leverage that discovery to compete with Microsoft without investors realizing that Google now agrees with Microsoft on the power of local applications.

The business and financial problem for Google is that the company has tied its financial worth to its identity as a web-based, modern replacement for the local-based, stodgy ideas of ordinary, local software vendors like Microsoft. Google cannot openly admit that the web doesn’t work for rich applications and that to deliver such rich applications it must get into the business of creating and providing local applications just like Microsoft. If it became evident that for growth into rich applications Google must, in effect, abandon the idea of time-shared sales of applications running on its own servers then perhaps even the most credulous Wall Street analyst might have to consider the complexities of competition with Microsoft in the local applications market, a fierce arena where one's talents with the sword must be sharp indeed.

The bottom line for the three-way Ajax story is that the only part of it that has distinctive meaning is the notion that web-based applications should be abandoned in favor of running local applications within the desktop machine if the objective is rich software. That's the entire point of the GIS and Networking essay and we are glad to see that Ajax partisans have taken the time to read that essay and to adopt its conclusions. But we are dismayed to see the conclusions of that essay used dishonestly in an attempt to fool investors, as the Ajax story is normally employed.
Some readers may object to our characterization of Java and XML as a sideshow or as "baubles" used to distract the natives with glittering trinkets. Such readers might fairly note that Java has evolved into a multi-billion dollar industry. That avoids the main point, as markets for languages can arise and thrive independently of the strategic consequences of using those languages to achieve dominance in operating systems and applications. Fighting and winning over applications depends upon what is done with the languages used and not upon the choice of programming language. Choosing a particular language no more guarantees success in applications than GM's choice of English units or Metric units for bolt sizes guarantees it success against the formidable quality and functionality of cars produced by Toyota.

We trust the discerning reader understands we do not in any way suggest that Ajax as a concept is a bad thing. On the contrary, except for the mindless embedment of JavaScript and XML, the whole idea behind Ajax is that the GIS and Networking essay is right and that rich applications should be run locally. Quibbling over minor details such as how the application is delivered or whether or not it fetches data from time to time from remote sources is missing the forest for the trees. If you support Ajax, whether you realize it or not you support the core conclusion of the GIS and Networking essay, that rich applications demand the distributed processing and rich bandwidth of local execution.

Likewise, we don't mean to criticize Google Earth in any way. "Earth" is a virtuoso application, an absolutely masterful implementation that any computer science professional will enjoy and cite as a distinctive accomplishment. But it is masterful precisely because it is a local application and for its core functioning abandons the idea of executing all that elegant code on some centralized server farm instead of within the user's local machine.

The dissection of Google Earth in this essay serves mainly to illuminate the difficult business task now facing Google as it expands into real competition with Microsoft, to balance the technical imperatives of casting more function into local applications with the business imperatives of not becoming seen as a local applications company by Wall Street. That is not an easy task even for business leaders with the proven skill of Google's management.

But while Google and Microsoft battle it out in the big leagues, users considering how and when to apply applications should not allow themselves to be fooled by misleading use of acronyms, like Ajax, into thinking that a local application is a web application or that web applications can magically avoid the issues set forth in the GIS and Networking topic.

The wise developer and investor will also not allow themselves to be led astray into wasting resources on activities that do not deal with the fundamental challenge, to deliver a high performance experience on the local desktop that cannot be outmatched by any competitor who knows better how to wield the power of distributed, local computing. Whether you call such efforts to harness local computing "Ajax" or whether you call it business as usual in the competitive world of Wintel applications, the task still remains to master all aspects of high performance local computing.

And for our users, when they are asked their opinion of Ajax we hope this essay will encourage the immediate response, "Yes, I think Ajax is great and that's why I use Manifold!"

See Also

GIS and Networking
Using RAM and other Machine Resources
Using RAM and other Machine Resources

Before reading this essay, please read the Performance Tips topic if you have not already done so.

Manifold System was designed for the modern era of desktop machines. How is the modern era different from ancient days?

- Random Access Memory (RAM) is cheap.
- Disk storage is free.
- Processors are cheap.
- Graphics is both fast and cheap.
- Local networks are fast.
- Humans are very, very expensive.

Machine Architecture

Let's take a moment for a quick refresher on key factors in machine performance. The fastest thing in your computer is the processor. Modern processors will run at well over a gigahertz in speed with certain internal parts of the processor running several times faster. Modern processors are so fast that they end up spending much of their time waiting for the slower parts of the computer system to catch up.

The second fastest thing is RAM, which usually operates at a third to a fourth of the speed of the key parts of the processor. The third fastest thing is the video subsystem, which usually consists of dedicated video RAM, a graphics processor and video circuitry that can read/display from video RAM at the same time that the graphics processor or main system processor can work with the video RAM. It is often said that the human retina is brain tissue that has been extended out into the eyeballs for the purpose of acquiring visual info. In a manner of speaking the RAM in video subsystems is like that in reverse: it is part of the processor - RAM ensemble that has been extruded out into a display subsystem for the purpose of projecting visual info.

A bottleneck in the high-speed part of your machine is the linkage between the processor and RAM. Processors are so much faster than ordinary RAM that it pays sometimes to install a bit of extra high speed RAM, a cache, so that for many memory interactions the processor need not be slowed by RAM. It also affects the speed and power of graphics cards that must exchange lots of data at high speed with the processor and system RAM.

[An aside: using graphics processors as massively parallel computing engines is an exception to the above simple case view. Such architectures are best understood as massively parallel supercomputers residing within the graphics card served by their own local RAM, which are connected to the "other" computer, that is, the CPU and its resources through a very fast parallel system bus connection. If you have one of those installed and can use it, the architecture of your computer is re-shuffled so that the fastest thing becomes the GPGPU and its very fast local RAM, assisted by a supporting cast of very fast CPU and its local RAM. The meaning of this essay still holds, because such ensembles are thousands or millions of times faster than anything else, in particular, faster than disks or local area networks.]

Compared to the three-way ensemble of processor, system RAM and graphics engine everything else in your computer system runs thousands of times slower. It’s quite literally the difference between a fast bullet and a slow walk. For example, the next fastest things in computer systems after processors and RAM are hard disk drives and the local network interface.

Hard disk drives and local networks are extremely slow compared to processors and RAM. A fast disk might have an average access time of a few milliseconds, which is over a thousand times slower than the microsecond access times typical of RAM. Although hard disks can transfer data faster once data is found they are nonetheless best understood as subsystems that are a thousand times slower than RAM.

Even hundred megabit or gigabit networks are also extremely slow compared to processors and RAM. At best, although the fastest purely local networks have the potential to go faster, even a good network interface will end up being about as fast as a local hard disk. This is one of the main uses of a fast network, to enable the dispersion of hard disk storage to different machines on a local network without any performance penalty attached to using a disk on a different machine. That this is accomplished is not a tribute to the speed of the network but rather an indication of how terribly slow hard disks are compared to the speed of processors and RAM.

An aside: Internet, by the way, is idiotically slow compared to even slow things like disks and networks. If the processor and RAM are like a fast bullet, and hard disks and local networks are like a slow walk by a human, then
even a fast Internet connection is like a snail. Suggesting that doing GIS over Internet is a better idea than doing GIS locally is like suggesting that snails are fast enough to outrun bullets.

For a software designer the main message of the above is that keeping the entire application in RAM so that it executes at processor and RAM speeds means we can make the application go a thousand times faster than if we have to use hard disk. That's a payoff too big to overlook in an era of cheap RAM.

For someone buying software and hardware it means the most important hardware investment is in the amount of RAM installed. Lots of RAM means that the processor will not be standing around waiting for the hard disk to provide paged data. More RAM is usually better than investing in a faster processor because without adequate RAM it doesn't matter if the processor is faster: both a fast and a slow processor will be spending most of their time waiting for slow fetches of paged memory from hard disk.

This effect, by the way, is one reason why Intel processors surged ahead of other processors for UNIX use in the late 1980's even though Intel 386 and 486 series processors were usually slower than the Motorola, SPARC or Alpha alternatives. The Intel processors could be purchased within PCs where RAM was much cheaper than RAM sold for use in the more expensive workstations that used the faster processors. PC clones loaded up with lots of inexpensive RAM were more effective UNIX platforms than fast workstations hobbled by small amounts of expensive RAM. The extra RAM made the PCs faster overall even though their processors were slower.

The same effect is repeating itself in the second decade of the new millennium, as yet another generation of computer users discovers that tossing lots of inexpensive RAM at a task in local processing makes life go really fast and pleasant. Once again, people are discovering through personal, bitter experience that if performance and a rich user experience are the goals, time-sharing over very slow networks is a terrible idea for anyone except the time-sharing salesman.

**Manifold Architecture**

The good news is that RAM is cheap. At the present writing, 12 GB of RAM (eight gigabytes) retails for under $200. At such low prices it is simply criminal not to have at least 4 GB of RAM. Manifold therefore expects that we will have lots of RAM in our machine. Occasionally, one reads criticism of Manifold that the system uses a lot of memory. If the memory is available, of course it does! It would be irresponsible not to do so. By using RAM whenever possible Manifold assures that it will run very fast. You must, of course, run 64-bit Windows to take advantage of lots of inexpensive memory. See the 32-bit and 64-bit Manifold Editions topic.

There is no point in slowing down the system so that it uses disk just in case the user has not invested in RAM. Such architectures penalize all users including those people smart enough to have spent a few pennies on RAM. To use this RAM, of course, it is important to have a modern Windows system such as Windows 7 x64 or Server 2008 x64 or XP x64 that does a good job of managing large amounts of system RAM. Investment in RAM and into modern Windows is very smart.

Manifold also expects we will have lots of hard disk space on our machine because disk space is now free. Perhaps it is not completely free but it is so close to being free that for all practical purposes it is free. For under $100 one can now purchase over a terabyte (one thousand gigabytes) of storage. At those prices inadequate disk space should no longer be a factor for most people.

In contrast to the low cost of RAM and the essentially free status of disk space the cost of educated humans has skyrocketed. Almost anything we can do from a hardware perspective is worth doing if it saves human time. This is one reason that Manifold now saves all components of a project within a single .map file. By keeping all data in a single file we eliminate any wasted time hunting about for different files that make up a project.

Finally, Manifold expects that one has a modern graphics card installed with serious hardware support for graphics, GPGPU capability and well-written graphics drivers. Very hot graphics cards with massive amounts of dedicated graphics memory are now commonplace and dirt-cheap. There is no reason to waste the time of a user with slow graphics when lightning-fast graphics is cheap. If we can utilize GPGPU computing (see the NVIDIA CUDA topic) it is crazy not to spend $100 or so for supercomputer performance in those functions.

**Good Business**

Manifold's business structure is designed to exploit the interplay of the above technical factors. The low price enables users to invest into modern Windows, lots of RAM, fast processors, fast graphics cards and large hard disks. These machine resources will serve you every day no matter what application you launch. Many users will already have modern machines in place and will need no investment at all.
The main difference between Manifold and other, non-GIS applications is that large GIS projects involving images will often be the very largest jobs in memory usage ever executed on a particular machine. GIS images, for example, can easily grow to tens of megabytes or even hundreds of megabytes. This is much, much larger than images typical of simple graphics arts work and a factor of ten or greater than even very large word processing documents. Adding RAM is therefore an expected step for serious GIS work involving large images.

Some readers are blessed with modern systems and will not have to worry about hardware resources. Others have to make do with less contemporary hardware. As time goes by and machines improve a 3 Gigahertz processor will be something so slow it is purchased only for children's machines. In that time a reference to a Core i7 will seem positively antique. However, there are many people in the world who will be using slower systems for years to come so we offer the following advice.

Some suggestions when assembling a new system:

- Just about any AMD or Intel dual core or quad core with 1 or 2 GB of RAM and Windows from XP to Windows 7 will be fine for working with small drawings in an ordinary vector GIS way. Databases and drawings are low-overhead items.
- When working with images, we suggest at least a quad core machine with 4 to 8 GB of RAM running Windows XP x64 or Windows 7 x64. This is excessive for small images but necessary for mid-sized images. In modern times, most people buying new machines will buy at least this much anyway. Images are computationally intensive.
- RAM has become ludicrously cheap. Many Intel Core i7 machines can host 12 GB of RAM. Buy lots and lots of RAM. It's dirt-cheap and it saves you time, which is very expensive.
- A fast graphics card must be installed when performing 3D rendering in terrain windows. NVIDIA-based cards currently selling for under $50 will provide smooth motion even with large terrains. We like cards using NVIDIA graphics processors and strongly suggest getting the hottest NVIDIA processor-based card you can afford. Using NVIDIA CUDA can be a huge performance advantage as well.

See also the advice in the Performance Tips topic.

As a reality check keep in mind that one can configure an incredibly hot machine and purchase Windows, Microsoft Office and Manifold System and still have thousands of dollars left over as compared to the cost of buying just the same GIS software capabilities from a legacy GIS vendor. On occasion, one encounters people who seem proud of having burned up many thousands of dollars per seat on legacy software that does less than Manifold. It is especially funny to realize that these are often the same people who complain about no longer having enough budget to afford modern machines, adequate RAM or even an upgrade to 64-bit Windows 7.

If people within your organization oppose increased Manifold procurement a good approach is a table in two columns that in one column lists the cost of legacy software required to approach Manifold's capabilities. In the other column one can present a list of all the wonderful modern hardware and software one can afford as a result of purchasing Manifold. The difference is real money that has a real impact on machine capability and organizational output.
Public Access to Public Data

manifold.net is committed to supporting public access to public data. We also support unrestricted use of cartography, which we see as a form of free speech. We believe in modern times that it is not possible for free people to freely participate in democracies unless they may also freely utilize digital maps.

This essay is not a rant against maps that are sold by commercial entities. People and companies should be able to create new maps with an expectation that they will be able to sell those maps for a profit. However, we think government maps and GIS data created by the government are a special case. If they are placed by law into the public domain they should be made available to the public.

Civic Life Depends upon Free Access and Usage

Many aspects of civic life are defined by geographic considerations or are profoundly influenced by the geographic context in which they occur. In modern times, the use of digital data by governments and societies has become so interwoven into politics and ordinary civic processes that the maps used to drive decisions and debates are digital ones. GIS software running on computers is required to view and to manipulate such digital maps. To an increasing degree, the image of the world that is used to govern its citizens is a purely digital representation.

If you are forced to use only paper maps, you are barred from seeing or working directly with the data sets that are often used to make decisions. Instead of being a participant in democratic decision-making, you become the recipient of decisions already reached. Note that the popularity of web-based "map servers" does not remedy this situation, since such web servers provide a picture of the map and not the actual map itself. With the actual GIS data one may be a participant; when provided with a report, one is merely a recipient.

If you are charged a fee for access to the maps upon which government decisions are based, you are being charged a poll tax on your right to participate as a citizen. Free maps are essential to the democratic process. This does not mean that all maps should be free of charge, but it does mean that the maps created and used by government should be free or nearly so.

A corollary is that government maps should not only be free of charge, they should be free of usage restrictions. If the government tells you what you can do with the map, it is the same as if they told you what you could say with your words. Perhaps there is no charge for speaking, but you may not speak freely. Free usage of maps extends to the right to freely reproduce them and otherwise to use them and to provide them to others as you see fit.

In the United States, GIS users have become accustomed to free access via Internet to a very wide range of maps. Manifold imports so many formats that one could spend a lifetime downloading and viewing maps and still not see all that are freely available. However, despite the immense richness of the US GIS data wonderland, it still lacks many key GIS data sets that are required for informed citizen democracy:

- It comes as a surprise to many technologically-literate people that even today, using the full resources of Internet, one cannot get a full set of digital maps of the world that are both usable by ordinary individuals and are free of usage restrictions. If NIMA’s VMAP1 were made fully-available, this major restriction would end.
- Until the publication of the first Free World Time Zones maps (by the Free World Maps Foundation), one could not even get free maps of time zones (!) for the United States and the world. There is still no official set of maps available to the public.
- Telephone Area Codes affect life and commerce and are a key aspect of telephone communications. They are heavily regulated by government and the subject of numerous laws and regulations, but one cannot get a map of telephone area codes that is both free of cost and free of usage restrictions.

In the United States there seem to be two contradictory trends in public access to public data. On the one hand, more public data than ever before is being published on the Internet for free download. On the other hand, many public agencies ignore laws guaranteeing public access to public data, or they are providing the data in a form that renders it unusable by the public. In general, there are good news trends and bad news trends. We believe GIS users have a civic duty to educate the public and their representatives in government to encourage the good news trends and to reverse the bad news trends.

Good News and Bad News

Recent trends provide good news and bad news for free people who wish to use modern maps:
Good News:

- Affordable GIS Hardware - Personal computers have become so powerful in recent years that anyone with an average PC can enjoy GIS capability with capacity and processor speed greater than the computers originally used to create most of the world's digital maps. As a practical matter, the cost of hardware is no longer a barrier for individuals in developed areas or for most organizations worldwide.
- Affordable GIS Software - Recent advances in GIS software, driven by the PC technology revolution, have made sophisticated GIS capability (like that in Manifold System) available at prices affordable by ordinary individuals.
- Expanding Public Domain data sets - Led by the United States, an increasing amount of government-originated data is emerging that by law is in the public domain and thus, in theory, is available for free usage.
- Internet as the Great Equalizer - Internet makes it possible for people around the world to cooperate as equals to promote public access to public data. The low cost of communications and data interchange over Internet makes it practical for maps to be created and exchanged using teams of people in different countries. Because it is technically difficult for governments to censor Internet interactions, even people in countries where free cartography is illegal can benefit from freedoms in other countries.

Bad News:

- State Cartographic Monopolies - Regrettably, the citizens of many so-called "free" societies must still purchase the landscape of their democratic discussions from a state-run monopoly and must utilize the maps so purchased only as allowed by the licenses granted by the state. Outside the United States, this is the rule, not the exception.
- Attacks on Cartographers - Repressive governments often restrict cartography by private citizens. Such restrictions are prompted not by an uneducated fear of cartography but by a calculating, politically informed awareness of the utility of cartography. Citizen use of electronic maps supports undesired activities such as self-assembly, free speech or economic competition with the government. Mercator himself was imprisoned after his travels for cartographic purposes triggered government suspicion. Some repressive regimes ban citizen cartography as a means of maintaining dictatorships. Other countries, including some so-called "free" Western democracies, make it illegal for private citizens to engage in cartography. The ban helps maintain the state-run cartographic monopoly.
- Repressive Influence on "Free" States - Some state cartographic monopolies have influenced agencies within free countries such as the United States in an attempt to prevent freely usable data from emerging that may threaten the state monopoly. In the US, for example, as of this writing the National Imagery and Mapping Agency (NIMA) has refused to release to the public many parts of "VMAP1", the public domain, non-classified, highly detailed world map it possesses, in part because of influences from state monopolies outside the US. The sharing of data between governments has now introduced the repressive policies of some state monopolies into formerly free areas. It also appears, in our opinion, to have become a handy excuse for rogue agencies to maintain an imperial disdain of statutory requirements to make public data available to the public.
- Denial of Public Access - Some agencies simply refuse to provide public access to data or restrict access to approved users. If it is truly public data, you have the right to access it. Some agencies will not release public data to ordinary individuals, researchers or others not meeting their criteria for an approved user, or they will attempt to place a restrictive license on data in the public domain. Some agencies will make chilling statements about the law that are flatly untrue. For example, the Census Bureau at one point published instructions for TIGER/Line recipients that told them TIGER/Line was a trademark of the Bureau and that any usage of TIGER/Line data must be accompanied with a notice to that effect. That's simply wrong, since one can use this public domain data set however one wishes without any need to publicize the Bureau's trademarks. In another case, the EPA used restrictive licensing to deny access for years to the general public to important public domain GIS data and code regarding chemical emergencies.
- Privatization by Public Agencies - Some agencies in the United States are privatizing public data and selling it back to the public as "products" using the lexicon and practices of private business. Although US law requires agencies to provide data for no more than the cost of copying and distribution, some agencies charge far more than allowed by law. The resources available to agencies are so great and the cost of utilizing the legal system in the US is so high that as a practical matter few people can afford to challenge even blatant violations of the law. Some agencies still charge thousands of dollars for data that costs tens of dollars to reproduce and distribute. In an even more disturbing trend, some agencies have registered trademarks on public domain data sets and have used their registration of the trademarks to attempt to prevent any "competition" from sources redistributing the public data for free.
- Privatization of Public Data by Private Firms - One increasingly hears of key public data sets becoming privatized through opaque dealings with private firms. For example, by law telecommunications and telephones policy such as tariffs are a matter of public record in the United States; however, it is not currently possible to discover a map as fundamental to telecommunications debate as is the map of telephone area codes. A private contractor currently holds this map as a privatized data set. If you do not have free and open access to such maps as a private citizen you will forever be kept at a disadvantage in public hearings on telecommunications policy. In another example, the Census Bureau once released a CD that stated a commercial GIS vendor's license (who was involved in the production)
controlled all data on the CD including the 1997 Statistical Abstract of the United States, an indisputably "public" data set. In 2001, the USGS effectively privatized the entire body of SDTS DEM files by delivering them in a de facto, exclusive deal to a small private company in Florida for distribution.

- Privatization through Private Influence on Federal Agencies - A subtle, but very real, means of privatization of public data occurs when market concentrations in GIS software causes governmental agencies to publish public data in proprietary, private formats that require the permission of private companies for people to fully use the data. For example, much Federal data is published in ESRI's undocumented .e00 format. Although firms like manifold.net can discover the contents of such formats, there are very few other firms who can do so. In addition, there is no guarantee of transparency of data as would exist if the data were provided in an openly published format such as ESRI's .shp format. Although agencies could publish the data in an openly published format such as ESRI's .shp format, they will nonetheless at times publish data only in .e00. At times it seems that some agencies have a promiscuous interest in releasing data only in deeply proprietary formats, such as ESRI "geodatabase" formats that not only are deeply proprietary but require the vendor's latest products so that even licensees of the vendor's earlier products cannot use such data. We feel this is a matter that education and awareness can help solve. However, it is a very important matter since to the degree any agency publishes data in a proprietary format it is not publishing data that is freely usable by the public.

- Illegal restrictions by indirect requirements of tools - Some agencies which are required by law to provide public data in open formats will nonetheless use proprietary formats where the only tools available enforce proprietary restrictions. For example, a common misconception is that .sid format is OK to use for public data because the makers of the format at Lizardtech (an appropriately reptilian name) provide a "free viewer" which can display the data and export it to open formats such as GeoTIFF. However, if you try to acquire and to use that "free viewer" you'll find that it is a licensed, proprietary tool which requires users to agree with Lizardtech's grossly objectionable legal positions as a legal requirement of using the tool. It is a bit like justifying illegal restrictions on your right to vote by saying that they are not a factor if you sign an agreement saying that you can vote, but only on the condition that you agree your voting rights can be taken away at will. Such restrictions may not be objectionable to some users, but they will be objectionable to competitors of the proprietary data format (for example, ECW or JPEG 2000, both available in open source code) which restricts the ability of a wider community to utilize that data for the convenience of end users.

- Privatization by Researchers - As open as scientists normally are, there are few communities as secretive as "Big Science" can be at times. Many images of our world are in the form of public domain maps and data created by NASA and other government Big Science initiatives; however, these can be surprisingly hard to get when researchers feel a need to "protect" data from access by an unknown public. Numerous scientists are very open about sharing data, so we certainly don't wish to suggest that science overall has become "closed". However, it is not fair to criticize government and private firms for privatization of public data without acknowledging that the scientific community has been remiss in this area as well. Researchers withhold public data from the public for a variety of good and bad reasons, but if the result is to deny public access to public data the result is bad for democracy.

- Privatization through "Webstacles" - A "webstacle" is the use of the Internet to avoid providing data to users. This is often the unintended result of good intentions, but at times webstacles are used to inappropriately privatize data. Here's how it works: an agency spends a few million dollars creating data. Rather than provide it for free download via FTP, they spend a few more million dollars creating a "map server" web site that spoon-feeds data to users in the form of dead images served over the web. The agency then says "we are fulfilling our duty of providing the data to the public through this web site." What's really going on is that the public cannot get free, public access to the data. The web server does not provide any data, only reports on the data in the form of images. If you don't like what the agency decided to show you in those images or if you wish to use the data in your own maps or analyses, tough luck. Further, webstacles traditionally only provide microscopic glimpses of the real data set. The agency has no risk that public users might ever actually get their hands on the entire data set. The result of the webstacle is that the data set remains inaccessible to the public. Map servers are a great idea if they are backed up with an FTP site or other means of making public data truly accessible to the public, but they cannot be considered a source for data if all they do is serve images.

We think on the balance it is possible to leverage the "good news" above to defeat the "bad news." When free citizens around the world join their GIS expertise via Internet, it only requires a single copy of a public domain dataset to emerge in order to defeat the "bad news" reported above. Likewise, it is possible to engage in cartography in those countries where cartography is still legal to defeat government restrictions on cartography in repressed societies.

What You Can Do

Here are specific actions you can take to promote public access to public data:

Appendices

4305
Learn about copyright laws and Federal and State laws providing public access to public data. If you know the law, you’ll know when someone is trying to restrict your rights under the law.

Learn about the GNU "Copyleft" license used to publish GNU and Linux and how it can be used to publish GIS data in a way that will prevent its subsequent privatization.

Learn how to use the FOIA and how to make and pursue FOIA requests for GIS data from Federal agencies.

Publish public domain maps and GIS data sets you obtain using the GNU license. Provide them to people who run servers allowing free download.

If you provide map images via Manifold’s web server capabilities, consider providing access to the underlying data via FTP as well. This is not always appropriate for commercial data, of course, but all public agencies providing public access to public data should attempt to do so.

Encourage authors of public domain cartographic data sets to publish data under the GNU license. Do not patronize vendors who attempt to privatize public data or who reprint public data at unreasonably high fees. Support commercial vendors who provide real value added or original data by purchasing their products.

When you encounter a federal or state web site that features webstacles instead of real data, contact the administrator of the site and ask them to include links to real data in downloadable form. Some agencies mean well: because their agency uses over-priced software they think that GIS costs many thousands of dollars per license so they put effort into providing images in the belief that the public cannot afford to work with the real data. Set them straight: tell them that Manifold makes it possible to work with the most sophisticated data to as great a degree of sophistication as one chooses at prices no higher than ordinary Microsoft Office applications. Tell them the more sophisticated the data is, the more important it is that users be able to work with it as real data.

File Freedom of Information Act requests with US governmental agencies (or the equivalent with State agencies) to obtain GIS data that is not provided to the public. If after asking politely for the real data behind a "webstacle" web site the agency will not give it to you, make an FOIA request for the data. Sooner or later the webmaster will realize that it is easier to provide a link on the site than to respond to a FOIA request. Very important: once you get data in response to a FOIA put that data into circulation so that other people can get it without repeating your FOIA request.

Educate federal, state, and local government worldwide on methods of publishing data to promote free map usage. This would include technical advice to avoid "privatization traps" such as publication in proprietary formats, as well as offers of free redistribution and access to Internet for organizations that do not have the technical resources or budget within their agencies to offer data that is being accumulated. Explain the difference between serving images and providing access to original data.

Complain to your Congressional representative and Senators about clearly illegal actions by federal agencies such as illegal FOIA responses and clearly illegal privatization of public data. Of course, unless you have given your Congressperson money in the form of political contributions you cannot expect any assistance but even in modern times there are surprisingly many Representatives and Senators who have directed their staffs to take an interest in legitimate complaints raised by "ordinary" people. Therefore, it is not totally an act of naivete to raise a complaint with one’s Congressperson. If your Congressperson ignores you (as they almost always do on matters that require any thought) and you are among the one tenth of one percent who can afford to use the US justice system, litigate the denial of your FOIA request.

If you choose to take legal action over illegal agency actions, such as illegal responses or denials to a FOIA request, seek out allies. Many legal firms will have a "pro bono" (for free) program of contributing legal work for public causes. Seek out organizational allies and other people who may be affected by the illegal action. For example, NIMA’s refusal to release the VMAP 1 coverages for Vietnam (clearly US origin data) is a slap in the face to the many US veterans who fought in Vietnam and who wish to use maps to preserve their history of the conflict, to note war graves and to seek closure in MIA/POW questions. Your Senator will likely ignore a single letter from you, but quite likely a series of letters arriving from Vietnam veterans and veterans’ organizations will nudge the Senator’s political instincts into action. Likewise, a law firm may be more likely to donate "pro bono" legal work if more than one person is involved.

If you are a government employee working within an agency that is deliberately restricting public access to public data (as, for example, NIMA does at this writing with VMAP?) make written notes of any such activities you observe. If you hear of any legal case filed, send your notes anonymously to the legal team filing the case. It will be a big help.

Encourage financial support from foundations, individuals, companies, and other organizations to promote public access to public data.

Tell your friends about manifold.net and encourage them to buy a copy. Manifold supports public access to public data (where else would you read a rant like this?) and provides a practical tool to get GIS data out of exotic Federal formats and into formats usable by many different GIS products. Manifold is also the only affordable way that ordinary people can work in a sophisticated way with sophisticated data.

The Freedom of Information Act (FOIA)
The Freedom of Information Act and its recent update, the Electronic Freedom of Information Act, guarantee access to virtually all GIS data created by the US government. There are only nine statutory exclusions to FOIA plus a pseudo-exclusion related to certain copyrighted map information provided to the Federal government. Use a good search engine to find sites devoted to FOIA and making FOIA requests. Numerous web sites will provide step-by-step instructions. These notes discuss issues especially relevant to GIS.

Many states have statutes equivalent to the FOIA for access to state information. Some, such as California's public records act, are considerably stronger than the FOIA because they award attorney's fees to the information requester if they have to go to court to enforce the act against an uncooperative agency.

Universal Access through FOIA

FOIA covers virtually everything created by the Federal government. If you see a printed map or a web image created by the Federal government, it's almost certain you have a right to the GIS data that was used to create that map or image. Ask politely to get the data. If you are ignored, send in a FOIA request letter. Send all letters by certified mail, return receipt requested so they know you know when they got the letter. The agency then has 10 or 20 days to respond.

If your FOIA request is denied, they have to tell you why, citing explicitly which exemption is being used. Don't accept any excuse that is not in the specific statutory exclusions. For example, NIMA stonewalled release of VMAP1 for years with excuses such as "our security office has not approved it" and so on. If it's not actually classified (a specific Federal designation that is the result of a specific process) it doesn't matter what their security office thinks of the matter. If it is not classified or not a copyrighted map provided by a third party under (rare) certain conditions you have a right to get it.

In recent years, some agencies have seeded public data with private, copyrighted material as a means of tainting all of the public data so that it cannot be forced into public hands via FOIA requests. We believe this is illegal and that the agency would have to provide, at a minimum, all data it originated, "unwinding" the poisoning of the data at its own expense. A related tactic would be to file a FOIA request for all of the documents the agency has that prove the copyrighted material is, in fact, copyrighted and did not enter the public domain as a result of a dealing with the Federal agency.

In many circumstances material provided to the Federal government enters the public domain automatically unless certain procedures are followed to keep it out of the public domain. This is to prevent contractors from selling data to the government for public use and then seeking to deny the public use of the data. If the supposedly "copyrighted" material used to taint the public data is also in the public domain then this FOIA exemption no longer applies.

Costs

Agencies can charge the cost of locating and duplicating information in response to a FOIA request, although the first two hours are free. You can ask them to process your request only up to the two free hours allotted by law. If you ask for electronic transmission of GIS data via FTP, the marginal cost of sending GIS data via FTP is zero. You should be able to get almost any GIS data set within the statutory two free hours. Agencies may attempt to deny you access by threatening to levy very high costs for what really costs nothing to FTP. Hold your ground and appeal all such constructive denials of your rights under FOIA. If you seek a large data set and do not have a large FTP site, contact the webmaster@manifold.net to inquire about access to our mammoth FTP sites for use as incoming sites to receive FTP delivery of data pursuant to a FOIA request.

Be Fair, Be Firm

It is essential to know FOIA and your rights under it. It also is critically important from a moral and tactical perspective not to abuse FOIA by filing requests for data that has already been published, or to use FOIA to harass your fellow citizens in government who are doing their best to make data available. Please be flexible and be reasonable whenever using FOIA. Most government agencies are highly professional in responding to FOIA requests. Understand the process and do careful research on the data you seek so you can write a focused, efficient FOIA request that does not require unnecessary work to process.

In many cases, making a few patient and responsible telephone calls or sending a few emails will eliminate the need to file a FOIA request. Many agencies and researchers are perfectly happy to provide a copy of a public data set in an informal manner. If someone does help you informally, do not abuse their goodwill by publishing their name on the Internet so they end up regretting their help to you. Do your best to help republish the information so that you too help disseminate the data.
Likewise, once a FOIA request is filed you should be firm in advancing the request. When asking for a politically sensitive data set you may be contacted by government representatives. They may try to convince you either not to pursue your FOIA request or to agree not to republish any data you get. Hold your ground. Be reasonable, but be firm. The requested material is either in the public domain or it is not. If it is not and the agency denies your request, they must cite the FOIA exemption used for denial. Get everything in writing and do not make verbal agreements authorizing delay, modification or abandonment of your request.

Most FOIA requests are handled in a highly professional (albeit slow) manner. If an agency has no political or internal "furf" issue with releasing data, you'll often get what you request very quickly. If the data requested is politically sensitive the situation is quite different. Our experience is that the art of using FOIA with unresponsive agencies is mainly the art of persevering in the face of various bureaucratic maneuvers intended to make you forget about your request.

It is important to persevere with FOIA even if you don't intend to (or cannot afford) the ultimate step of going to court to compel action on your FOIA request. Filing a FOIA request and then following through is a simple matter of sending a few letters. If enough people file FOIA requests for important data sets held hostage by recalcitrant agencies (as, for example, in the case of VMAP1 being withheld by NIMA), then pressure builds upon the agency to deal fairly. Within many people file FOIA requests for key data sets, sooner or later the agency faces a real showdown in front of a judge where the judge will see a case record littered with numerous examples of underhanded agency maneuvers. In such cases judges have been known to award millions of dollars to FOIA requestors to compensate them for their legal costs incurred.

The GNU License

The Free Software Foundation originated the GNU ("GNU is Not UNIX") project to create a freely distributable version of UNIX. One of the results is Linux. To support this effort the Foundation created a license under which software could be freely distributed and freely used.

As used by GNU, the usage of the term "free" above is not focused on a meaning of "free of cost". The key notion is freedom of usage. Many data are free of cost but suffer drastic restrictions on usage. For example, GIS companies often provide maps that are free of cost but which are subject to severe restrictions on utilization such as restriction to use only with the proprietary software packages of the originating GIS company. Such restrictions make it impossible for such maps to be used in open public policy discussions, because the restrictions require everyone participating in the discussion who wishes to see the full richness of the data in the maps to agree to the various proprietary licensing restrictions of a particular GIS company.

In other cases, non-profit organizations often publish maps that are provided free of cost but which may not be used for commercial purposes or by commercial organizations. We strongly believe that speech that is edited, even for a "good" purpose, is still not free speech. For someone to say you may use a map but not for commercial purposes as a practical matter is a very significant intrusion into your organizational life. It raises the question of when your behavior is "commercial" and when it is not.

As anyone knows who has done accounting for organizations that engage both in "commercial" and in "non-profit" activities, the distinction is often quite difficult to make and highly subjective in nature. Note also that blanket prohibitions on "commercial" usage would prevent the "chain reaction" mechanism of shareware publication that has enabled millions of CDs to be published at very low prices and near-zero cost to software authors. Commercial interests can often be harnessed to publish data and software at very low cost for the public good.

It has often been remarked that "public domain" data is already in the public domain and thus constitutes a freely distributable data set already. The problem with "public domain" data is that anyone may change such data slightly or convert it into a new format and then claim a copyright on the new form and use such a copyright claim to restrict further distribution. In fact, much of the GIS map data now sold by private companies under onerous restrictions originated in "public domain" data. Perversely, much data has emerged "in the public domain" only to be intercepted by a web of proprietary interests before it can reach users.

The GNU "copyleft" license resolves all the above problems by copyrighting the underlying programs or data and by licensing them under a specific license. The license (in informal summary, see the license itself for the precise terms and conditions) says you may freely use the data in whatever way you wish, so long as you pass on in any subsequent redistribution or usage the clearly stated right for any of your users to do the same. This goes for any modifications of the data or programs covered by copyleft. You can change the data, edit it, improve it, simplify it, sell it, or give it away. However, you must clearly inform all of your users that they are free to do the same as well so long as they too, in turn, agree to the full provisions of the GNU copyleft license.

In this way, any maps that are distributed under the GNU license become seeds for yet further redistribution. Any improvements to the maps likewise become available to anyone else. Because of the GNU copyleft license, no one can take your maps and use them as a means of restricting usage, conversation, duplication, conversion to any other format, or any other restriction. This does not prevent anyone from selling programs or other data that
may accompany GNU-licensed maps. It simply means one cannot restrict such maps in any way except by the "chain reaction" terms of the GNU license itself. It is this chain reaction effect that has been an important factor in the worldwide growth of software like Linux.

We don't suggest that the GNU license is the only way to license maps for distribution. We simply recommend it as one way of publishing GIS data in a way that will assure continued public access and free usage if that is the intent of the publisher. There are many other reasons for creating and publishing maps other than to provide public access to public data. In particular, private access to private data is an entirely different thing.

Please do not read this essay to infer in any way that manifold.net is against commercial map publication for profit. In our view, respect for private property rights is the flip side of the same coin as respect for public property rights. Just as public data should be protected against privatization, private data should be protected against forced transition to a de facto public domain status. Private vendors have every right to offer their intellectual property for sale (including maps) however they see fit. Private map creation and sale for profit are essential mechanisms for the creation of many important, value-added or original new maps.

Acknowledgement

Much of the material in this topic is used by permission and comes verbatim from the writings of the Free World Maps Foundation, which seeks to assure public access to public data in support of citizen democracy.
Zip Codes are Not Areas

ZIP codes are postal codes in the United States created by the US Postal Service. Perhaps the most common misconception in GIS is that Zip codes are polygonal regions or areas. People often think of mapping in the US as a hierarchy of ever-subdivided polygonal areas: states, counties, cities, zip codes. If they need higher resolution than a county, they next leap to zip codes because they think of zip codes as polygons. This is not true.

Zip codes are linear features associated with specific roads or with specific addresses such as apartment buildings or military bases that are best regarded as a point. In some cases, Zip codes have no physical location because they are assigned to a mobile or abstract “location” such as a military ship.

Even in the most common case of Zip codes assigned to streets, Zip codes do not clump together in groups that may be covered by rational polygons. We can consider an example using a map of part of Reno, Nevada, shown below. This map is fairly typical of the situation in mid-sized urban areas. It is extracted from the US Census Bureau’s TIGER/Line 1997 data set, which includes roads as segments of lines, with most line segments coded with Zip and Zip+4 codes for that particular segment. In this note, we will refer to both Zip codes and the Zip+4 extension together under the name “Zip code”.

To create polygons from road lines where lines have a common zip code there are several approaches. One possible approach is to select all line segments with the same Zip code and to then draw an area (polygon) that encloses them. This can be done by creating a buffer zone about each street line having a particular Zip code and then doing a Union of the buffer zone areas thus created. The blue, purple and green areas were created in this way and each represent a a different Zip code value.

The road lines shown in red selection color all have yet another Zip code in common. Immediately there are three pathologies visible in this map.

First, note that the blue area is not contiguous. Second, note that there are many regions of overlap between the blue and the purple areas and between the purple and green areas (we should have used varying layer opacity so that the regions of overlap were clearer). Third, note that at least one road segment highlighted in red (all having the same Zip code) occurs inside the purple zone where it is completely surrounded by all adjacent streets having a different Zip code.

The above situation is extremely common. In fact, we selected this particular map at random. Any urban map in the US will show similar, if not even more bizarre effects. Rural maps can have such a sparse network of roads
with such strange zip code assignments that some rural areas cannot even be approximated with zip code regions.

For the above reasons, any map that purports to show “Zip Code Areas” or “Zip Code Polygons” should not be taken as a precise map showing zip code locations. It is at best some sort of approximation and most likely is wildly inaccurate in certain regions. The approximations can be useful, but they should not be confused with the real thing.

The US Postal Service, of course, doesn't make it any easier to deal with such issues by making it easy to get Zip code information. Zip code information is not available for download via Internet from the US Postal Service. It is best obtained from (of all agencies!) the US Bureau of the Census.

**ZIP Code Tabulation Areas (ZCTAs)**

For statistical tabulation purposes the Census Bureau has long found it convenient to work with Zip code groupings of population. Zip codes have been so useful that the Bureau embarked on a project to create a standardized map of the US showing the approximate region of coverage of various Zip codes as areas. These areas are known as **ZIP Code Tabulation Areas (ZCTAs)**. ZCTAs may be downloaded from the Census Bureau’s [www.census.gov](http://www.census.gov) site. Drill down to the Cartographic Boundaries pages to get ZCTAs. Download them using .e00 format so they will import into Manifold using the correct NAD83 datum.

Before ZCTAs were published, every vendor of maps used in GIS had to resolve the various ambiguities posed by Zip code pathologies like those shown above. With ZCTAs the GIS industry can now use a standard approximation that is the same used as the Census Bureau for publishing demographic information. It is not clear if the Bureau will continue to create ZCTAs after the year 2000; however, they are so useful we believe they will become the industry standard for maps representing Zip codes as areas.
Just Say No to GIFs

If you run a web site please try to avoid using .gif format. Use .png for images that must be lossless and use .jpeg for imagery where lossy compression is OK. It's not just good technology it's also a matter of ethics.

Here's why:

The LZW (Lempel-Ziv-Welch) algorithm for compression in software has been patented not once but several times: It has been patented by both IBM (U.S. patent 4,814,746) and Unisys (U.S. patent 4,558,302). British Telecom has also been awarded a similar patent in the UK. IBM's patent was filed three weeks before the Unisys application. Both patents were granted in 1983. In a typically inept maneuver the U.S. patent office issued patents to both IBM and Unisys for the same algorithm, apparently not realizing that the patents covered the same "invention."

.gif was developed by CompuServe and published as an "open" format in the 1980's without realizing that the LZW compression used to create compressed .gif files could be subject to patent claims by Unisys or the other LZW patent holders. Based on the alleged "open" nature of .gif it became a widely used format. At some point, Unisys became aware that .gif files were created using LZW and decided to begin "enforcing" its LZW patent. Unisys has sought to extract money payments from companies who create software that writes .gif files, in particular in 1994 and 1995 launching a series of initiatives that successfully alienated numerous software developers. At one point, Unisys claimed to be able to require royalty payments from every web site that had a .gif image in it. In contrast, IBM has never sought to enforce its LZW patent against .gif developers and users.

In 1995 CompuServe launched the GIF 24 initiative, to develop a replacement for .gif not held hostage by software patents. Other developers launched the GEF initiative, which soon converged with GIF 24 to create .png. .png is a true "open" format not subject to patent claims by anyone. It is more efficient than .gif to boot and a much better choice for network graphics in the modern age.

Many developers and patent attorneys feel that it is only an accident caused by the ineptness of the U.S. patent office that Unisys was awarded a patent on LZW compression in the first place. Many developers feel so badly about Unisys' campaign that they have pledged to maintain only "gif free" web sites and avoid purchasing Unisys products or services. See http://burnallgifs.org for an example of some developer feedback on Unisys and .gif.

At manifold.net we now use .png format instead of .gif for all graphics publication on web sites. We have done so mainly because .png provides better compression and thus better performing web sites for lower bandwidth. Even so, it's also nice to know that with every .gif replaced with a .png we help strike one more blow against wrongheaded patent policies.

Remember: friends don't let friends drive drunk, and they don't let them use .gif either!
Beware of Counterfeit Software

By some estimates, over 40% of the software used in business settings in the United States is illegally copied or "pirated." Many businesses and users in the US will use copies of software for which they are not licensed. In some countries, well over 95% of the software in use is pirated. Software pirates include casual, amateur copiers of software as well as those who sell and those who buy professionally pirated software products.

In most Western countries, the number of users who make casual copies of software in violation of their license agreements greatly exceeds those who use the products of professional pirates. At manifold.net we feel an important part of keeping prices down is to provide a level playing field so that those customers who play by the rules do not end up subsidizing those who cheat. Using serial numbers and activation effectively eliminates casual cheating, the bulk of piracy in most Western countries. We feel that strong criminal sanctions by government authorities will make trafficking in professionally pirated products too risky for most users in industrialized countries.

In some countries virtually all of the software products that are sold are professionally pirated products that are ripped off on a massive, industrial level. It is always possible that professional pirates may circumvent the serial number and activation mechanisms in Manifold System. However, the technical effort to do so is very substantial and immediately marks any vendor or possessor of such a pirated form of Manifold System as a blatant lawbreaker.

Governments in countries where piracy is rampant have begun to realize that widespread piracy prevents the development of an indigenous software infrastructure within their countries. No one will create software for a market where software is universally ripped off. As a result, those governments have begun cracking down on software pirates.

All genuine Manifold System software from Release 5.00 onward requires a serial number and activation for permanent installation. If you have been supplied with Manifold System 5.00 or a more recent edition that does not require a serial number and activation, you may have been the victim of a counterfeiter. Contact sales@manifold.net to find out if the license offered to you is a valid license.

Resist the temptation to purchase or to install counterfeit software. When visiting countries in which counterfeit software is openly sold it is tempting to purchase a CD for a few dollars that contains thousands of dollars of software packages. However, when returning through Customs with pirated software you risk severe legal consequences if you are caught, including possible detention, conviction and the life-long stigma of a felony criminal record.

Customs officers know to examine hard disks and are very good at finding counterfeit software goods when people return from countries where such goods are sold. Remember, Customs officers look tens of thousands of people in the eye and do searches for hours every day of their work week for years on end so they have much more experience detecting a possible problem than the novice smuggler has at concealing it. You may think you have the iron nerves required to bluff your way past an experienced Customs officer, but when you face that officer across an examination table and realize that he or she can strip search you or otherwise detain you it is too late to realize the stakes are not worth the risk.

Criminal penalties are often much higher than the amateur smuggler realizes. For example, when smuggling a counterfeit CD into the US it is very easy for the dollar value of the ripped-off software on the CD to total many thousands of dollars, which escalates the offense into the range of a serious felony. Even if extended jail time is avoided upon conviction, in the US people with a felony conviction lose many rights for life, including the right to vote, the right to own firearms and even the right to work at many jobs or professions. That is true whether you are caught in Customs or caught when pirated software, unknown to you, calls down authorities within your own country.

If offered counterfeit software within your own country or via the Internet, consider that counterfeit editions of popular software packages may include viruses or Trojan Horse "back door" programs that later allow entry into your computer system. Such programs can be used to automatically harvest (via network connections) information from any computer system running the Trojan Horse. This is a classic way to take over machines to steal credit card information or identity or to use your machine to perpetrate credit card theft or to generate spam. Criminals can use Trojan Horse programs to use your machine to make purchases on their behalf using stolen credit cards. When the crime is discovered, it will be your computer and your Internet connection that will seem to have committed the crime.

Criminals use counterfeit software to distribute viruses and Trojan Horses for three reasons: First, virtually all software requires Administrator rights to install so the criminal knows you will be giving his installation script full Administrative permission to do whatever it wants in your machine. Second, embedding malicious code within counterfeit software is a great way to avoid scrutiny by anti-virus or malware packages. Finally, the criminal
knows that when you install counterfeit software you yourself will be committing a crime so you won't seek help from law enforcement authorities in case of trouble.

A final consideration is that some software companies are taking active measures against counterfeiters by placing special lures within their anti-piracy code that they know will be found and eliminated by counterfeiters. However, such lures can be craftily designed so that when they are removed the application begins emitting signals through Internet connections that advertise the use of a pirated copy to the software company and to law enforcement authorities. By using what you think is a safely "cracked" counterfeit copy, you could be inviting the police to raid your home or business.

Although it is true that raids are rare, especially in countries where piracy flourishes, even in the worst countries the authorities from time to time like to crush a few pirates, especially end users who are not making payoffs and won't be missed anyway, to show they take the piracy problem "seriously." In such cases only a fool would paint a target on themselves by unwarily using a cracked copy to invite a raid.

If you have never considered using counterfeit software the above comments obviously do not apply to you. However, you may find yourself some day counseling a friend or business colleague against the risks of using counterfeit software. The above information may help someone realize that the risks and possible costs of using counterfeit software are greater than they might appear at first glance.

Help Links

The topics in the Help Links chapter are here to capture and redirect users who may be searching the Help index using different terminology than that which is used within Manifold. These also provide indexing topics that provide a link to the main topic for that subject.

These topics provide cross-indexed links to the Manifold topics that cover the same subject matter.
**Arcs**

"Arcs" are called *lines* in Manifold.

**See:**

For Experienced GIS Users
Drawings
Terminology in GIS

**Note**

This topic is here to capture GIS users who are searching the Help index looking for familiar words. Use the links above to jump to the equivalent Manifold topics.
Polygons
"Polygons" are called areas in Manifold.

See:

For Experienced GIS Users
Drawings
Terminology in GIS

Note

This topic is here to capture GIS users who are searching the Help index looking for familiar words. Use the links above to jump to the equivalent Manifold topics.
Coverages

In vector-oriented legacy GIS products, the word "coverage" usually refers to a vector layer in a map that may be composed of vector objects such as points, lines and areas. Such vector layers are drawings in Manifold. Because Manifold maps can have drawings, images or labels components as layers in maps, we do not use the word "coverage" within Manifold. We simply speak of layers or we refer to the specific component type and speak of drawings, images, or labels layers.

Note that sometimes the word "coverage" is used in a slightly different context to mean "category of layers". For example, the US National Imagery and Mapping Administration (NIMA) uses this word within the VMap data set series to refer not just to one layer but to an ensemble of layers with a particular category. A boundary "coverage" within VMap data sets, for instance, may consist of several sets of line, point or area objects that in the normal use of the word would be considered separate "coverages."

See:

For Experienced GIS Users

Maps
Layers
Drawings
Images
Labels
Terminology in GIS

Note

This topic is here to capture GIS users who are searching the Help index looking for familiar words. Use the links above to jump to the equivalent Manifold topics.
**Grids**

The word **Grid** may occur in three contexts in Manifold:

- To import data sets from legacy GIS products, where surfaces are known as "grids."
- To create a mesh of horizontal and vertical lines, like a graticule.
- To denote a type of projection specific to some country.

In vector-oriented legacy GIS products, the word "Grid" usually refers to a regularly arranged data set that is used to show elevation or other data. Such a data set may be either a raster or vector data set. In Manifold, such data sets are referred to as **Surfaces**. See the Surfaces and Terrains topic for information. **.e00 Grid Files** is a choice in the File - Import - Surface dialog.

The word **Grid** in Manifold is used to refer to an orthogonal mesh of vertical and horizontal lines, like a graticule. See the View - Grid topic for information on creating grids.

Within the projections supported by Manifold are many "National Grid" projections. These are usually adaptations of standard projection types with parameters specific to a particular country.

**Importing ESRI ArcInfo exported raster grid files**

**.e00** is the preferred format for importing grid files from ESRI products. However, ArcInfo can export raster data using "gridscii" format either as integer or floating point data. There appear to be no standard extensions for such files in universal use within the ESRI community. We suggest saving (or renaming) such files to use either a **.grd** extension for integer data or an **.flt** extension for floating point data. Use the File - Import - Surface importer set to **GRD** files for integer data and **FLT** files for floating point data.

**See:**

For Experienced GIS Users
- Surfaces and Terrains
- Importing Surfaces
- Manifold Projections
- National Grids
- View - Grid

**Note**

This topic is here to capture GIS users who are searching the Help index looking for familiar words. Use the links above to jump to the equivalent Manifold topics.
**Thematic Mapping**

Thematic mapping is called Thematic Formatting in Manifold. To create thematic maps we use drawings that are linked to tables that have data fields for the different objects in the drawing. We then use thematic formatting to change the color or other display characteristics of the objects based on a data field.

To create a thematic map we need to know how to import a drawing, how to link a table to it if it does not have a table already linked to it when it is imported and how to create a map using that drawing. We then need to know how to launch and operate the thematic formatting dialog.

**See:**

- Drawings
- Formatting Drawings
- Thematic Formatting
- Tables

**Note**

This topic is here to capture GIS users who are searching the Help index looking for familiar words. Use the links above to jump to the equivalent Manifold topics.
**Georeferencing**

Georeferencing is called **georegistration** in Manifold. See the Georegistration topic.

**Note**

This topic is here to capture GIS users who are searching the Help index looking for familiar words. Use the link above to jump to the equivalent Manifold topics.
Digitization
Digitization is called **tracing** in Manifold. See the Tracing topic.

**Note**

This topic is here to capture GIS users who are searching the Help index looking for familiar words. Use the link above to jump to the equivalent Manifold topics.
Digitizing Tablets
Manifold does not use digitizing tablets. The preferred method of working with images is to use scanners to scan paper maps as images and to then use Manifold's tracing tools and image editing tools to work with those images. Because the computer allows us to zoom far into images we can work with the mouse within Manifold to much greater accuracy than is possible with a digitizing tablet. Scanners are much less expensive than digitizers, are much more accurate and the scanner - mouse - software ensemble is much more flexible.

See the note at the end of the tracing topic for a discussion of digitizing tablets.

See

Tracing
Editing Images
Georegistration

Note

This topic is here to capture GIS users who are searching the Help index looking for familiar words. Use the links above to jump to the equivalent Manifold topics.
Map Algebra
Manifold has a free-form formulas and calculation capability that works on surfaces. This capability is available if the optional Surface Tools extension is installed.

See

Surface Tools
Transforming Surfaces
Transform Dialog Functions and Operators

Note

This topic is here to capture GIS users who are searching the Help index looking for familiar words. Use the links above to jump to the equivalent Manifold topics.
Making Mosaics
We can create mosaics of images or surfaces by copying and paste or paste append.

Suppose we have a set of images we would like to combine into a single image. Duplicate one of the images and give that duplicate a name, such as *my mosaic image*, and open that image. Open each of the other images in turn, hit **CTRL-A** to select all of the pixels in the image, **CTRL-C** to Copy all of the selected pixels, click onto *my mosaic image* to make it the active window and then Paste Append to paste the pixels into the mosaic image. As you paste append each image in turn into the receiving image it will expand as necessary. The end result is a single image that has pixels pasted from all of the other images, that is a mosaic of the other images.

The general technique, applicable to just about any georeferenced images and surfaces is set forth in the Download and Mosaic TerraServer Images topic.

**Note**

This topic is here to capture GIS users who are searching the Help index looking for familiar words. Use the links above to jump to the equivalent Manifold topics.
Rasterize

Converting a vector drawing or a map using vector drawings into an image is called rasterization in many GIS systems. See Printing to Files or Images.

Note

This topic is here to capture GIS users who are searching the Help index looking for familiar words. Use the links above to jump to the equivalent Manifold topics.
Slope and Aspect from Elevation Data

Surfaces can automatically show slope and aspect. Import a surface, open it and use View - Display Options to choose slope or aspect.

Note

This topic is here to capture GIS users who are searching the Help index looking for familiar words. Use the links above to jump to the equivalent Manifold topics.
Table - Transform Toolbar and Operators
See the following topics:

Transform Toolbar - Tables
Transform Operators - Tables
Transform - Using Tokens and Text Strings
Transform - Span Excluding / Including

Note
This topic is here to redirect GIS users who are searching the Help index looking for familiar words. Use the links above to jump to the equivalent Manifold topics.
Thiessen Polygons
What are known in some GIS circles as "Thiessen" polygons are known to mathematicians and to most computer users worldwide as Voronoi cells. Manifold uses the more prevalent term.

See:

Transform - Voronoi Operators

Note
This topic is here to capture GIS users who are searching the Help index looking for familiar words. Use the links above to jump to the equivalent Manifold topics.
Vectorizing
Vectorizing is called tracing in Manifold. See the Tracing topic.

Note
This topic is here to capture GIS users who are searching the Help index looking for familiar words. Use the links above to jump to the equivalent Manifold topics.
Warping Images
Warping images to fit a desired projection or to match another image or drawing is called georegistration in Manifold. See the Georegistration topic.

Note
This topic is here to capture GIS users who are searching the Help index looking for familiar words. Use the link above to jump to the equivalent Manifold topics.

What's New
What's New
This topic provides a starting point for users of previous Manifold System releases to see highlights of what is new in this release.

Users new to Manifold may safely ignore these topics and simply proceed with learning the system beginning with the Welcome and Read Me First topics.

Manifold System Release 8 introduces next-generation performance infrastructure for Manifold System products with faster rendering, extended spatial DBMS support and introduction of massively parallel processing for supercomputer performance in surface transforms. This new infrastructure enables hundreds of improvements in Release 8 and will provide the foundation for thousands of improvements in future updates and releases.

Although there are few overt changes in the user interface, the impact of infrastructure improvements, such as providing dramatically faster rendering, completely changes the feel of the product in daily use in a way that appeals to both beginners and experts alike. No one who has experienced 8 will want to go back to 7x.

Release 8 also responds to the increasing maturity and sophistication of the Manifold user community. New features supporting spatial DBMS in a revolutionarily inexpensive and transparent way open the door to individual users to employ spatial DBMS in their daily work. Spatial DBMS now costs no more than a single Enterprise Edition license.

Some individual users might think that spatial DBMS is not for them, that it is something that only concerns enterprises. Nothing could be farther from the truth - spatial DBMS is for everyone who wants lightning fast performance with large images and surfaces as well as fast and reliable storage for drawings. Modern DBMS is easy to install (installing an Express edition of SQL Server 2005 Express takes but minutes) and performance gains are dramatic.

Those same spatial DBMS features have become essential to small organizations moving into GIS as a result of the new affordability of GIS Manifold makes possible. Even if an organization has only three or four people doing GIS they still can benefit from features, such as multi-user editing, that previously only very large organizations could afford. Large organizations will also benefit from fully orthogonal and flexible spatial DBMS, so they run legacy technologies such as SDE side-by-side with modern replacements such as Microsoft SQL Server 2008 spatial technology.

Release 8 was developed around several focus areas:

- The prime focus was on next generation Manifold rendering engines for substantially faster display speed. Performance has been improved through re-engineering of rendering pipelines coupled with hundreds of tweaks throughout the system. The most visible aspect of the new rendering technology is progressive rendering, which frees the console for continued work even as a large display is rendered in portions.
- A second focus area for Release 8 has been dramatically extended support for spatial DBMS, whether using native spatial DBMS products such as IBM DB2, Microsoft SQL Server 2008 spatial (codenamed "Katmai") and PostgreSQL or using Manifold spatial DBMS technology that now confers spatial DBMS capability upon virtually any DBMS to allow storage of drawings, images and surfaces. In addition, manifold.net has released the new Manifold Spatial Extender for SQL Server as a free download to provide high performance spatial index capability for SQL Server 2005 installations until SQL Server 2008 becomes available to the public.
- The third focus area for Release 8 was transparent support for ESRI "SDE" geodatabases and ESRI "Personal" geodatabases in full read/write/edit capability using Manifold to interact with such databases.
Appendices

- A fourth focus area was extension of Manifold’s ability to use multi-core and multi-processor hardware into the supercomputer performance world of NVIDIA CUDA. Manifold can now take advantage of NVIDIA CUDA equipped systems to execute many functions virtually instantaneously using as many as 512 stream processors via massively parallel algorithms.

- A fifth focus area was extension of scripting capabilities into user interface scripting, allowing programmers to utilize virtually any element of the Manifold user interface within scripts. This allows extremely rapid prototyping, development of scripting "macros", fast implementation of "one-off" tasks and even creation of automated demo programs that manipulate the user interface under programmatic control.

In addition to the above focus areas, Release 8 provides the usual Manifold implementation of a few hundred user community wishlist items. These range from support for IronPython as a .NET scripting language to routine updates such as support for the latest TIGER/Line format. Beta testers have especially praised some new innovations, such as the new Autohide / Popup Panes capability, added in response to user community requests.

**Transition Issues for 8**

Key operational differences between Release 8 and the predecessor Release 7x include:

- Release 8 can read prior .map files but earlier Manifold release such as 7x cannot read Release 8 .map files.
- Release uses a new .map file format that enables much faster opening of images and surfaces. When an earlier .map file containing surfaces or images is opened, the format will be automatically upgraded to Release 8 format. This process will require some time and patience, but it is a one-time process. Afterwards, the new format .map files containing surfaces or images will open dramatically faster than before.
- Release 8 allows changing the area of interest for drawings linked from a DBMS. Such facilities will not work for linked drawings created with prior Manifold builds. Delete the drawing and re-link it.
- Applications using the Manifold API will almost certainly need to be recompiled. Programmers should study the detailed release notes for a guide to changes in code that may be necessary.

**Highlights**

Highlights of Release 8 include:

- **Faster .map file loading** - The new .map file format and enhanced data structures allows much faster opening and loading of .map files, dramatically faster in the case of projects containing large images and surfaces. In some cases, what used to take tens of minutes to open now opens in a few seconds.
- **Faster image and surface rendering** - Images and surfaces are rendered much faster, typically one hundred times faster in the case of very large images and surfaces. What used to take tens of minutes now happens in seconds.
- **Faster drawing rendering** - Drawings also are rendered much faster, often two to ten times faster with large drawings.
- **Progressive rendering** - Manifold now renders components using progressive rendering, where the user interface becomes active and available even before the component finishes rendering. This provides much faster workflow, because if we are opening a component with the intent of zooming into some part of it we can draw the zoom box and begin zooming as soon as enough of the component has been rendered for us to get our bearings.
- **Faster selections** - Visual selection tools have been optimized to work faster, typically two to four times faster.
- **Extensive View Caching** - Views are now cached so that Manifold rarely needs to re-render a view once it has been displayed. Using the Back and Forward view navigation buttons, for example, is instantaneous.
- **NVIDIA CUDA support** - Manifold will automatically utilize NVIDIA CUDA hardware that may be installed in the system and supported with CUDA libraries. Currently supported in many surface transform dialog functions, CUDA-capable plug-in cards provide as many as 512 stream processors for supercomputer power on the desktop for as little as $150 or less. Some applications fitting CUDA capabilities that might have taken tens of seconds or even minutes can be accomplished in hundredths of seconds. Although limited in the current implementation, CUDA is the first step in what the months ahead will clearly emerge to be the future of almost all GIS computing. In terms of revolutionary steps forward, this could well be the Release 8 feature with by far the most impact upon the future of GIS.
- Autohides / Popup Panes - We can now expand the size of working windows by auto-hiding panes, such as the project pane. Move the mouse cursor over the pane's tab and it dynamically expands for use. Move the mouse away and the pane hides itself into the tab to allow a larger working view.

- Manifold Spatial DBMS capability storing drawings, images and surfaces within almost any DBMS - Manifold Enterprise Edition or higher now can confer spatial DBMS capability onto almost any DBMS, including those that do not have native "spatial" facilities. Manifold can automatically create spatial indices to store drawings using a variety of supported geometry types and can also store images and surfaces within high-performace spatial DBMS storage. Drawings of virtually limitless size can be edited using Area of Interest specification on desktop machines. Manifold can store images of virtually limitless capacity with very high performance access and rendering using almost any commercial or open source DBMS you choose, allowing opening of even multi-gigabyte images in seconds. Even if DBMS is not used to store drawings, the ability to store images with very high performance into DBMS is something all serious users of Manifold should learn to exploit.

- Enterprise Edition support for Spatial DBMS export - Previously, Database Administrator Edition was required for convenient export of drawing into a spatial DBMS. With Release 8 any Enterprise Edition or higher edition can easily export drawings, images or surfaces into spatial DBMS in a single step, automatically creating spatial indices, marking the spatial table to be treated as a drawing and otherwise managing the spatial DBMS infrastructure to allow subsequent use of drawings.

- All editions can utilize Manifold spatial DBMS capability - Enterprise Edition continues to be required to connect to native spatial DBMS (DB2 with IBM Spatial Extender, Oracle OCI, PostgreSQL/PostGIS, SQL Server 2008) using native spatial capabilities, but any Manifold edition from Personal Edition on up can connect and utilize generic spatial DBMS capability conferred by Manifold once Enterprise Edition has been used to upload drawings, images or surfaces into the spatial DBMS. That includes use of spatial DBMS within SQL Server 2005 using the Manifold Spatial Extender for SQL Server.

- Manifold Spatial Extender for SQL Server - The spatial extender provides high performance spatial indices within SQL Server 2005 or more recent version. Works with SQL Server 2005 Express SP2 or more recent edition as well. Use the spatial extender with SQL Server 2005 to get started today with spatial DBMS within the SQL Server world. Manifold works automatically and transparently with the spatial extender when it is installed on a SQL Server machine for better performance.

- Spatial DBMS support for Microsoft SQL Server 2008 Spatial - This new Microsoft product (code named "Katmai") has been announced but has not yet been published for the general public. If you are lucky enough to have a pre-release version, your copy of Manifold has built-in support for SQL Server 2008 Spatial.

- Spatial DBMS support for IBM DB2 with IBM Spatial Extender for DB2 - Manifold also adds DB2 to the roster of native spatial DBMS products supported by Manifold. The IBM Spatial Extender for DB2 is a free download for IBM DB2 Express-C editions as well as full DB2.

- Spatial DBMS support for PostgreSQL - Breaking with the prior tradition of supporting only commercial DBMS products, Manifold now provides full native spatial DBMS operation with PostgreSQL equipped with the PostGIS spatial extension.

- Spatial DBMS support for ESRI SDE geodatabases and ESRI Personal geodatabases - Manifold can connect to SDE (also known as "ArcGIS") data stores using any DBMS supported by ESRI or to so-called "Personal" geodatabases, most frequently encountered within Access .mdb files. Manifold can import drawings from such geodatabases or link to drawings for read/write/edit dynamic compatibility with such linked drawings to add/delete/edit objects in such drawings, even changing their projections.

- Faster data access through the Data Source dialog - Access to data sources such as OLE DB, ADO .NET and ODBC has been streamlined into a new Data Source dialog. The new dialog remembers data sources that have been configured and makes them available within dialogs such as the Database Console or Administrator console. This makes it much easier to connect to databases because once a data source has been configured we can quickly double-click on it in the Data Sources dialog to use it.

- Changeable Area of Interest - Drawings linked from any spatial DBMS using an area of interest subset can now have the area of interest changed even after they are linked.

- User Interface Scripting - A massive new capability that enables programmatic control of virtually all Manifold user interface elements and even many non-Manifold user interface elements as well. Instead of working exclusively through objects within the API, programmers can now control the Manifold user interface as if someone were automatically calling commands from menus, clicking control boxes, making choices, entering text and so forth. All Manifold dialogs (many hundreds of dialogs) have been adjusted to allow user interface scripting. Uniformly praised by beta testers as a revolutionary step forward in convenience, power and flexibility.

- Numerous scripting improvements and new objects - The scripting system has been enhanced with new options, such as forcing a script to run in a separate thread if desired. Many new objects provide access to new Manifold features.

- Expanded command line and logging options - A variety of new command line options support discovery of programmatic command names as well as a set of new logging options that enable automatic recording of timings such as rendering and execution times.
• **Choice of IMS rendering formats** - The Internet Map Server can now render using a variety of graphics formats, such as .jpg or .gif in addition to the default .png format, as well as allowing specification of rendering quality.

• **IronPython scripting support** - We can now use IronPython, the outstanding .NET implementation of Python supported by Microsoft, for scripting Manifold in addition to the other .NET and ActiveX languages supported.

• **Extensive additions to imports and exporters** - Numerous new capabilities have been added to importers and exporters, such as a NetCDF importer and numerous improvements in TIFF (16-bit LAB data and 32-bit/64-bit CMYK data), DXF/DWG, improvements in KML/KMZ exports (layer transparency, extended coordinate accuracy, etc.), yet more options for PRJ file recognition and improvements to other formats.

• **Coordinate system (projection) improvements** - Numerous improvements have been added, ranging from the ability to better support rotation factors in custom datums to numerous new datums for European countries. EPSG support has been integrated to allow use of Microsoft SQL Server 2008 spatial, and extensive work has been done on auto-matching of coordinate systems with spatial DBMS in PostgreSQL and DB2 spatial extender.

• **Faster geocoder handling of extension address errors** - Misspelled addresses geocoded against a geocoding extension database can now be processed twenty times faster.

• **Numerous other improvements and bug fixes** - Release notes for Release 8 have been changed to separately itemize changes in the main development path from bug fixes within new features or features continued from 7x. Current release notes cite a total of 415 new items and bug fixes. Some new items, such as much faster image rendering, are massive features which themselves incorporate hundreds of internal changes. User interface scripting alone required thousands of discrete changes within the code base.

---

**New Help Topics in 8**

- DataSource
- DataSourceColumnSet
- Developing Applications
- Drawing - Area of Interest
- ESRI Geodatabases
- Example: Configuring SQL Server 2008
- Example: Linking a Drawing from SQL Server 2008
- Example: Storing a Drawing in Manifold Spatial DBMS
- Example: Storing a Drawing in SQL Server 2008
- Example: Storing a Surface in Manifold Spatial DBMS
- Example: Storing an Image in Manifold Spatial DBMS
- Example: Storing an Image in SQL Server 2008
- Example: Tracing Virtual Earth into SQL Server 2008
- ExportDb2
- ExportPostgreSQL
- ExportSqlServer
- ImportDb2
- ImportNetCdf
- ImportPostgreSQL
- ImportSqlServer
- Manifold Spatial DBMS Facilities
- Manifold Spatial Extender for SQL Server
- MapServerRenderFormat
- NVIDIA CUDA
- RecentDataSource
- RecentDataSourceSet
- Show Area of a Parcel in Acres
- Spatial DBMS
- Spatial DBMS Facilities
- SQL Server Spatial DBMS Facilities
- The Data Source Dialog
Info

Limitations

Although Manifold may be used on Microsoft Windows 98, there are several key limitations in this operating system. Most limitations will also apply in Windows Millennium (Windows Me). To avoid such limitations, use Windows Server 2003, Windows XP or, in a pinch, Windows 2000. Manifold System is no longer supported in Windows 95 or Windows NT.

The best environment for Manifold System is Windows Vista, Windows Server 2003 or Windows XP. Windows 2000 is a fine system but visibly aging, so it is wise to jump to Vista, XP or Server 2003. Users are warned that support for Windows 98 and Me will not likely continue very long and may be discontinued at any time.

If this documentation is still in use at a time when support for Windows 98 and Me has been discontinued by manifold.net, all references to Windows 98 or Windows Me should be disregarded.

Windows 98 Limitations:

Areas not resizable - Area styles are not resizable in Windows 98 or Millennium. Area styles are fixed at size 5 in those systems since all they support is an 8 x 8 pixel bitmap matrix. See Area Styles and Size.

Crashes in Windows 98 - Manifold uses standard Windows virtual memory management facilities when working with components too large to fit into available RAM. Windows 98 memory management facilities are unreliable for large projects and will crash when Windows Server 2003, Windows XP or Windows 2000 will not. In addition, printing through Windows 98 is less reliable because these systems do such a poor job of memory management that internal bugs in printer drivers are more likely to surface. These are not Manifold crashes but are operating system crashes. When working with large projects, use Windows Server 2003 or Windows XP.

Printing Problems in 98 and Me - Windows 98 and Me are full of bugs that often affect printing. We suggest using Windows Server 2003 or Windows XP to prevent bugs in Windows from interfering with print jobs. Windows 98 or Me are so buggy that tech support will only investigate reports of printing problems in Windows Server 2003, Windows XP or 2000.

Map Server Limitations

Manifold IMS (Internet Map Server) is published for use in Windows Server 2003, Windows XP (Professional and up) or Windows 2000 (Server or Professional running IIS). The map server has been observed to work using the personal Internet server in Windows 98 and Me; however, Windows 98 and Windows Me operating systems are not supported for use with the Manifold map server. Performance with 98 or Me will be seriously reduced as compared to Windows Server 2003, Windows XP or 2000.

General Limitations:

Maximum Size of Projects - There is no limit to the maximum size of Manifold projects, nor is there any limit to the number of components in a project.
**Maximum Component Size** - No single drawing, table, image or surface can exceed 16 EB (exabytes) in size. Since one exabyte is $2^{64}$ bytes, about a trillion gigabytes, as a practical matter there is no size limit for these components. One will run out of disk space before the size limit is approached.

Comments, queries and scripts are limited to approximately 1 GB. Labels are limited to 16 EB when initially created, but any changes made to a labels component after it is initially created cannot exceed approximately 1 GB.

Limits on comments, queries, scripts and labels component sizes are approximate because the precise limit depends upon details of the specific Windows system in use and the operation of that system. For example, in certain systems the 1 GB approximate limit on such components can expand to 2 GB or even 3 GB on machines run with a /3GB boot option. However, because such expanded uses depend on many complex interactions the stated limit size for comments, queries, scripts and changes in labels components is only 1 GB (a vast amount for such things).

Note that the 1 GB limit on "queries" is the size of the actual SQL query text, not on the results of a query. This is somewhat of an absurd "limit" since it is very rare that one writes an SQL query longer than a few hundred lines, let alone more than a megabyte in size.

**Maximum Process Size** - 32-bit Windows operating systems cannot allocate more than 2 GB of memory per process. This puts an upper limit on system functioning whereby algorithms that need more than 1 GB of memory (even temporarily) cannot usually operate. For example, using `Dissolve` to dissolve millions of polygons with very large data attributes for each record can exceed the process limit. Research continues at manifold.net to work around this limit of 32-bit Windows systems. As a solution until either 64-bit Windows systems become generally available or until Manifold research finds a way to exceed the limit imposed by 32-bit systems, try running large algorithms in parts. For example, try running a large `Dissolve` on portions of a drawing and then do a `Dissolve` on the results of these runs.

**Printer Limitations** - Manifold uses standard Windows facilities to print to printers. Manifold passes the print job to Windows, which then calls the printer driver. There is no contact between Manifold and the printer driver at all, so that any Manifold print job passed by Windows to the printer driver is guaranteed "legal" since it arrives from Windows using internal Windows system routines. Experience with previous editions of Manifold showed that numerous printer drivers do not fully support the requirements of a printer driver documented by Microsoft, so that some "legal" but complex jobs passed by Windows to the printer would crash the driver.

Manifold therefore uses a very simple set of Windows calls in an attempt to avoid triggering bugs in printer drivers provided by the printer manufacturer. If you experience difficulty printing, make sure to check the printer manufacturer's web site to see if there are more recent drivers available for your use, since the difficulty may arise from a bug in the printer driver that has since been fixed. Note that Manifold is sold for use with printers that fully support Windows requirements for printer drivers.

**Memory Requirements for Large Maps** - Manifold is designed for an era where RAM memory is cheap and personnel costs are high. `Undo` and may other user-friendly features require a lot of memory to implement. In order to make it possible to cancel out of certain operations, operating memory requirements may temporarily increase to three or four times the size of the map loaded. Normally, virtual memory will handle such requirements; however, for best performance we recommend installing lots of RAM.

**Free Disk Space Required** - Manifold requires up to four times as much free disk space as the size of an uncompressed project. This free disk space is required for temporary files that are created for project management, to allow `Undo` of some commands and to allow abandoning edits.

See the Performance Tips topic for more suggestions on improving system performance.

See the Memory Requirements topic for RAM and hard disk memory requirements.

**Software Requirements**

**OpenGL Required for Terrains** - Manifold requires a functioning OpenGL subsystem to display terrains. If there are no OpenGL capabilities in the system terrain windows will be blank when opened.

**.NET Framework 2.00 Required** - Microsoft's .NET Framework 2.00 must be installed to run this edition of Manifold System.
Process Debug Manager Required for Debugger - To run the Manifold Debugger you must therefore have the Microsoft Process Debug Manager (PDM) DLL installed on your system. See the Debugger topic for more information.

A Note on Operating Systems - The Manifold team recommends Microsoft Windows Server 2003 or Windows XP. We believe that Windows 98 and Me were outstanding products for their era. In general we would like to make sure our product works the same in Windows 98 and Me as they do with the latest generation of Microsoft Windows releases. However, since modern Windows versions provide more capability than do Windows 98 and Me we cannot in our product add capability that does not exist in Windows 98 or Me. Our support plan for Windows 98 and Me is limited to verifying that our code conforms to the Microsoft standards and that any errata are a result of limitations in Windows 98 or Windows Me.

As good as Windows 98 and Me may be, Windows Server 2003 and Windows XP are spectacularly better - even on "small" machines all of your software will run better and faster. This is not intended as a criticism of earlier Microsoft Windows products; rather, it is intended to acknowledge the major product improvements Microsoft has delivered over the years. Note that these comments especially apply when working with relatively large amounts of memory. We urge you to switch to Windows Server 2003 or Windows XP. It's worth it!

We no longer support Windows 95 or Windows NT since Microsoft itself no longer supports these versions with key systems features, such as .NET Framework 2.0, that are used by Manifold System.
Contacting manifold.net

You can reach the Manifold team by visiting our web site at www.manifold.net or by sending email to sales@manifold.net. Please consult the web site for telephone numbers and other contact information for sales and distribution channels in your area. The web site is the first place to look for all contact information or changes.

Technical support for Manifold comes through the same distribution channel through which you purchased your copy. If you purchased directly from manifold.net, see the Technical Support topic for information.

Please address all corporate correspondence for Manifold Net via email to sales@manifold.net.

No Response to Postal Mail

manifold.net does not respond to hard copy postal mail as the costs of using hard copy mail are very much higher than using email. Use email to contact manifold.net. Be green - go paperless!

Suggestions for New Features

Manifold users are strongly encouraged to send in suggestions for new features or alterations in existing features. Every new Manifold release and service pack includes numerous additions that have been prompted by user suggestions.

Ground Rules for Suggestions

Send email suggestions to sales@manifold.net. Due to the volume of suggestions received, suggestions for new features will not usually receive a reply; however, all letters conforming to the rules below are noted and their suggestions are tallied with others received and compared to current internal product plans. There are several rules that apply:

• Do not send any suggestion or comment that you do not wish to become a part of the public domain, that is, freely usable by anyone without any control by you or compensation to you. All suggestions or ideas received by manifold.net enter the public domain and are not confidential. Do not send any information or communication to manifold.net that you want to remain your intellectual property. If you send it to manifold.net you are giving permission for that idea or suggestion to be used any way that manifold.net desires, including free publication onto the web for anyone else to use and also including sale to a third party, without any compensation to you or intellectual property claim by you. If you work for someone else, do not send any ideas, information or suggestions that do not belong to you or that are not already in the public domain.

• Suggestions for new features are generally prioritized by how many people request them. For example, if more people request enhancements in image editing than request enhancements in CAD-style drawing editing, the image editing tools will more likely appear sooner. However, suggestions made by just a single person could (and frequently will) appear in the very next service pack if the product planning team at manifold.net believes they will serve a large user base or if in some cases they are very easy to do and fit into an available engineering slot that would otherwise go unused.

• Not all suggestions will receive replies, but at times you may be contacted by manifold.net personnel to discuss your suggestion. If so, any such communications are not a guarantee that your suggestion will be implemented. Do not base your purchase decision of additional Manifold licenses on an expectation that a suggestion will be implemented as a feature.

• Any new features for Manifold are prioritized based on company plans and trends perceived from user suggestions sent in by email. Whether or not a new feature is implemented in a particular service pack or release depends upon its priority, the amount of time necessary to implement it and the availability of engineering slots within the development organization. Sometimes features of relatively low priority will be implemented ahead of high priority features because the low priority feature is easy to do and an engineering slot opens up in which that feature can be implemented.

• The product planning process excludes form letters, bulk letters, letters that were obviously not composed as a personal letter to sales and any letters cc’d to third parties.

• For those suggestions that are contributed by only a few people the personal credibility and expertise of the suggester count for a lot. People who provide expert, complete and detailed suggestions win the credibility that makes it easier for product planners to commit more time to exploring the possible implementation of the suggestion, even if only one person has requested it.

Tips for Effective Suggestions
Some tips on how to make product suggestions that will influence the product as rapidly as possible:

- Don’t call a new feature you would like to see a "bugfix." For example, comments of the form "I've found a terrible bug in Manifold - it doesn't include a full-featured word processor like Microsoft Word" are a good way to lose credibility. If the product does not do something as it is documented, that's a "bug". If it does not include a feature or capability you would like to see, or if it is missing opportunities for additions here or there that greatly expand the usage of an existing feature, that's a new feature suggestion.

- Describe the new feature clearly and focus on specific features. The more detail you can provide about how it would work the better. General suggestions are appreciated but are not as likely to result in quick results. Example: "I'd like to see some more CAD editing tools" is too general to be very helpful. "I'd like to see move, scale, rotate, and flip commands like those that were in Manifold 4.50" is very specific.

- If enumerating a list of related capabilities, give some sense of your priorities. Example: "I'd like to see some more object construction commands. The most important in priority order are interactive Bezier curve drawing, the ability to draw a line perpendicular to another line or area boundary, and, if possible, a "centerline" line drawing tool that would constrain the drawing of a line to the mid-point of any two lines or area boundaries."

- The product planning process values independent votes for similar features much more greatly than a cluster of votes promoted by an advocate. If many people independently throughout the world come to a particular conclusion and find a new feature important enough to write, that tells us it is something that is really important enough for many people independently to decide to suggest. In contrast, if someone makes a posting on a newsgroup that says "I think feature X should always have default setting Y... write to manifold if you agree" and we get a few dozen letters saying "I agree" with a copy of the posting, that tells us the default setting was really not a big enough deal for more than one person to feel strongly about, because no one else noticed it or cared until the issue was raised in a new group.

- The product planning process values personal emails. Letters that are cc’d to third parties are excluded. Send emails to sales@manifold.net and not to the personal email accounts of any Manifold employees you may know. You may think you are lobbying the right person, but what actually happens is that employees will not vote for you by composing a letter on your behalf to sales@manifold.net and so your advocacy will not be influential.

- The product planning process does not allow time for planners to do your research for you. For example, at times writers will send a brief summary of what they are interested in and will append a comment such as "For details, see the discussion in..." appending a URL to a discussion thread in some Internet forum. Product planners will not run down that URL and attempt to parse a discussion thread to try to figure out what, exactly, it is you are advocating. If the issue is not important enough for you to provide a crisp, detailed and standalone suggestion of what you want, then your comment will carry little weight and will have much less impact then an effectively advocated suggestion.

- If you are new to the product, don't jump to the assumption that something cannot be done and so a new feature is required. Research the documentation and Knowledge Base carefully. Take a moment to try out your ideas with your peers, perhaps by posting to Manifold-L the GeoReference forum (see the Help - Manifold on the Web topic for fast access to the Manifold online community and other resources) to see if you've missed something already in Manifold. If some preliminary exploration shows the idea is not already accomplished, please send it in. If your idea is already in Manifold but in your opinion too hard to find, let us know about that as well.

Suggestions for New Formats

We frequently receive requests to add a new format to Manifold System for import or export. We are always willing to consider adding new formats to Manifold. There are four main determinants as to whether a new format will be added:

- Do we have full and complete technical documentation on the format? If we have publicly available documentation (such as a URL) that completely defines the format it becomes much easier to consider. Send such a URL or document to increase the chances your request will be implemented. Please do not send confidential information.

- Is the format a difficult one or an easy one to implement? Easy formats that are well documented will almost always be added rapidly.

- Have we had other requests for the format? If we have had very many customer requests it is more likely we will add a format even if it is difficult to implement. For an easy format that is well documented we have in the past added formats after only a single request.

- Are samples available of data in that format? The more complex the format, the more samples will be required to make sure the importer works correctly. If you send in a URL to a page that has plenty of samples the engineering team will know that any new importer can be tested. If there is no way to test the importer it is unlikely engineering resources will be committed to a complex new format.
Therefore, the best way to get a new format added to Manifold System is to send Manifold a URL or other document that precisely describes the format in full and complete technical detail and that provides links to sample data that may be freely downloaded.

Users sometimes write "Can you add XyzGIS format? Their home page is www.xyzgis.com." This is not helpful because unless a very large number of requests for that format are received manifold.net staff will not take the time to search the site for documents that might define the format. The betting at manifold.net is that if the requester hasn’t bothered to search the site then the desire for the format cannot be very great. If you would like to advocate a new format, you significantly increase your chances by doing the detective work to find a solid technical description of the format and by providing some samples.

Many of the formats in Manifold were added at the request of a single advocate who patiently located the required information and forwarded it to the Manifold team for implementation.

Requests for Broken Formats or Nonstandard Implementations

Manifold users will occasionally encounter files in formats that Manifold imports where the file does not accurately utilize the format. For example, there are many programs that can be used to write "shapefiles" which do not correctly write shapefile format, and there are images said to be in "GeoSPOT" format that are not written in accurate GeoSPOT format. Manifold.net will at times receive requests to alter the Manifold importers so they will read such pseudo-standard files.

There are two conflicting philosophies about staying true to a format definition. One philosophy says if something can be extracted from a file even though the format definition is not followed then do it. The other philosophy says that it is important to observe formats strictly and accurately as a protection against unknowingly importing damaged data. The first philosophy is often more convenient in the short term, but can lead to catastrophes in the long term. The second philosophy is more safe and professional, but can be less convenient in the short term.

Manifold tends to follow the second philosophy, of taking format definitions seriously. However, if there is a large body of data that systematically violates a format (for example, a large collection of data made available by a government), the manifold.net team is willing to bend the format definition a bit and allow such data. If you need such a relaxation of standards, the best way to argue your case is to provide a URL or other information that documents the existence of a large body of data in that "broken" format.

The above advice also applies to requests for nonstandard implementations of other standards. For example, a few GPS receivers say they use NMEA protocols when in fact they use a blend of standard NMEA with proprietary exceptions to NMEA. A request to support such "nonstandard standards" is a request to support something outside the standard and it will only be considered if there is significant usage that departs from NMEA in that way. For example, if a single, obscure GPS model departs from the NMEA standard there is little likelihood that the nonstandard implementation will be supported. On the other hand, if a major government organization buys a few million such GPS units then it is more likely that the nonstandard implementation will be supported.

Suggestions for New Projections

New projections are like new formats in that they are often easy to add if complete technical information is provided. The best way to get a new projection added to Manifold System is to send Manifold a URL or other document that precisely describes the projection in full and complete technical detail, including all formulae involved in defining the projection if the projection is not a simple parameterized variation on an existing projection. See the comments regarding suggestions for new formats as a guide.

Projections that are simple parameterized variations of existing projections are easily added to Manifold via customization. If a customized projection is indeed a standard projection in your part of the world it is an almost sure-fire candidate for addition to Manifold System as a "built-in" projection. If you would like such a customized projection to be added as a "built-in" within Manifold, send your customization XML as well as technical references (so the manifold.net team can confirm your work) as part of the suggestion. If you do not have a customization XML but you know that the projection is a variation on an existing one, please send a technical reference URL that sets forth the projection.

Frequently Asked Questions

Does sending a suggestion count as a support incident? Not unless it is embedded within a support request or a bug report to which a reply is requested. Example: "How do I thematically format a drawing using a text field in the sort order I desire?" Please add a feature to specify sort order." This will be treated as a request for support and will cost a support incident. Another example: "I need to draw Bezier curves and I think you should add this
Manifold® System Release 8.00 User Manual

capability to Manifold. Is it in there already and if not, how long will it take you to add it?" This is also a question of tech support and will use up an incident. It is a frequent occurrence that new users will ask a question of tech support about a "new" feature they want without realizing it is already in Manifold. Whether a feature is or is not in Manifold, if you ask any question of tech support you will use an incident. To avoid using an incident, make it clear you do not want a response of any kind. See the Technical Support topic for details.

**Why are cc'd letters ignored?** Cc'd letters are ignored for three reasons: 1) Cc'd letters that involve a third party will frequently involve a political dimension as people sometimes want to "show off" to their peers or try to convince others of their views. Just as in tech support, we feel communications are kept most clear if the temptations of playing to a crowd are removed. 2) At times, a good response to a suggestion may be a series of questions or other email conversation. We would like to conduct such conversations in private to allow manifold.net personnel to discuss usage with the person making the suggestion. We feel it is important to observe our privacy policy at all times, even if one of the participants is willing to waive it. 3) It takes time sometimes to parse a cc'd document that was not originally intended as a suggestion. If a direct communications pipeline to the people who create Manifold is not important enough for someone to craft a letter specific for that purpose, then we feel the content of that letter is not important enough to include in the planning process.

**What if my continued use of Manifold is dependent upon a new feature I have suggested?** You should never acquire licenses based on anything other than the capabilities of the released product. Manifold is always subject to revision and alteration as manifold.net judges best. Although the product's evolution is guided mostly by user demands, keep in mind that other users may have different priorities than you do so that the evolution of the product may not include suggestions you have made.

**How do I find out the status of a suggestion?** Once a suggestion is made there is no way to tell the status until it appears in a new release. Manifold.net is grateful for all suggestions but is unable to provide reports of whether or not specific suggestions have been adopted and if so, when they will appear in the product.

**How do I find out the release dates for new Manifold releases?** In general, these are not published. Users participating in Manifold-L, the discussion list for Manifold System users, or the GeoReference forum, the Manifold forum for the Manifold System User Group, will often hear rumors from factory personnel about general time frames in which new releases are to be expected. However, because manifold.net does not publish firm dates for new releases any such rumors should be treated as no more than broad predictions that are always subject to change. See the Help - Manifold on the Web topic for fast access to the Manifold online community and other resources.

**Why are not suggestions on Manifold-L or GeoReference monitored?** - The main reason is that all Manifold users can write directly to sales@manifold.net but only a minority of Manifold users participate in Manifold-L or on other online forums. By having one mechanism for intake, that of email sent to sales, we can assure all users equal status in making suggestions and guiding the evolution of Manifold. This policy also reserves for Manifold-L and GeoReference the freedom of having informal conversations and "thinking out loud" before sending in a suggestion. Quite a few suggestions for "new features" will turn out to be simply a lack of experience and not realizing those capabilities are already in the product. Testing out such ideas in Manifold-L or GeoReference is a good way for new users to benefit from the knowledge of their more experienced colleagues.

**I've seen extensive discussions of features in Manifold-L or GeoReference that surely resulted in lots of suggestions but appear not to have had any effect - why is that?** - First, recall that only direct communications to manifold.net count in the product planning process. Because surprisingly few users who participate in a discussion on an Internet forum ever bother to send a suggestion to manifold.net it is often the case that an extensive discussion yields few, if any, suggestions. Second, on those occasions when participants in a discussion do take the time to send a suggestion to manifold.net, quite often the suggestion is clearly unoriginal. Letters that are simple forwards of a posting with an "I agree" appended, are obviously prompted by an advocate, and so carry little weight, if any, in the planning process. Third, discussions in Internet forums are often very general in nature and provide little specific guidance as to new commands or features that are desired. Finally, only a small percentage of Manifold users participate in Internet forums, so it could be that a larger majority of other users has steered priorities elsewhere.

**Suppose I absolutely need a new feature - can I pay for it?** Yes, although the cost is usually very significant, in the several tens of thousands of dollars to the hundreds of thousands of dollars, depending on the level of custom engineering. Custom versions of Manifold System or scripts can be created with specific features that are required. However, given the propensity of "tire kickers" to waste resources on projects that have no hope of coming to fruition, a non-refundable fee of several thousand dollars is charged to review a proposal for custom engineering. This fee is charged because a significant amount of engineering effort is required to simply analyze a proposal for custom modifications and to determine the feasibility of such modifications. Contact sales@manifold.net with your requirements if you are ready and willing to pay the expenses involved.

If you have any suggestions to improve the above guidelines, please send them to sales@manifold.net.
About Manifold System

The manifold.net team first came together in 1993 to create an improved set of mathematics libraries for the massively parallel supercomputer (then the fastest in the world) that was a joint project between Intel and the US Department of Defense. Improved algorithms led to dramatic performance increases, in some cases resulting in two orders of magnitude improvement. The team later created a series of graph theory and computational geometry libraries as well as a series of "visual workbenches" used to teach the libraries.

The workbenches allowed visual selection of complex data, such as road networks, followed by the application of algorithms drawn from the programming libraries. The workbench products quickly evolved into a GIS-like visual network package that became Manifold System versions 1.0 and 2.0 in 1997. Clients who saw the package suggested the product was superior in many ways to old-fashioned GIS products and urged the company to enter the GIS market with products designed specifically for GIS users.

Manifold System first burst upon the GIS world in January of 1998 as Release 3.00. Within five months Release 4.00 was issued in June 1998, followed by Release 4.50 in March 1999. Release 4.50 incorporated several hundred suggestions from users of Release 4.00.

Release 5.00 was created between October 1999 and December 2001 with final release of Professional Edition in January 2002. 5.00 benefited from thousands of suggestions submitted by users of 4.50, as well as from numerous suggestions contributed by users of Manifold Database Commander and Manifold 3D View Studio. Suggestions came from all continents and numerous countries with especially heavy participation from the US, Western Europe, Russia, South Africa and Australia.

Work on Enterprise Edition paralleled work on Professional Edition with final release in July of 2002. Both Professional and Enterprise are based on a common code base so that a single executable can be launched either as Professional Edition or as Enterprise Edition.

Release 5.50 was already in progress during the Enterprise Edition campaign with all company engineering resources devoted to the 5.50 campaign beginning in July of 2002. 5.50 introduced unlimited sized projects, exabyte sized images and surfaces and increased the maximum size of other components to 4 gigabytes. In addition, hundreds of modules were re-coded for faster performance using larger component sizes. Hundreds of internal and external improvements were added for 5.50 as well, with 5.50 issued in January, 2003.

Service Pack 1 (SP1) for 5.50 introduced over 110 new features such as rotatable point styles and thematic formatting for labels in June, 2003, and Service Pack 2 (SP2) added hundreds more in October, 2003, including the next generation Manifold spatial SQL engine. In late 2003 the Business Tools, Geocoding Tools and Surface Tools extensions were introduced.

Issued in May, 2004, the 6.00 release included over 470 items in the initial release and then 320 items in 6.00 SP1 issued in October, 2004. Release 6.00 was the first Manifold release to require Microsoft’s .NET Framework.

Release 6.50, with over 620 items in the release notes, was issued in August of 2005 and benefited from many thousands of comments and suggestions by what has become a vast, worldwide Manifold user community. Beta testers in Asia, Australia, New Zealand, Africa, Europe, North America and South America participated in the largest beta trials ever for a new Manifold release. The result was clearly the best Manifold System release ever.

In November of 2005 the planned service pack for 6.50, called 6.50 SP1 was extended with plans to incorporate native support for Oracle spatial technologies. The collaboration with Oracle went so well that numerous features originally planned for future releases were pulled forward into 6.50 SP1. In early 2006 plans were extended yet again to cover yet more new features and to enable support for IBM DB2 as well as Oracle and SQL Server. After the number of features went over 500 items it was decided to issue this release as a major new numbered release, not a service pack, and so Release 7.00 went to press in May of 2006.

Release 7.00 introduced major new capabilities, including massive support for Enterprise class applications requiring simultaneous editing by potentially thousands of people of drawings that can be terabytes in size when stored on Oracle spatial servers. New Database Administrator and License Server products made it easy for large organizations to deploy hundreds or thousands of licenses, while a new Personal Edition product kept price low and features breathtakingly high for individual users.

Following soon after 7.00, with the release of 7x Manifold was been extended to run natively on x64 platforms. 64-bit editions of 7x allowed storing more data in RAM which, together with other enhancements to 7x, increased the capacity of Manifold operations and provide performance benefits. 7x also started an era of more frequent updates, made possible by improvements in internal processes used by Manifold engineering, that can be communicated in an automated fashion using a new Automated Updates option.
In 2007 Manifold 8 was released with numerous advancements. Release 8 extended native spatial DBMS support to IBM DB2 Spatial Extender, PostgreSQL and SQL Server 2008 spatial. Release 8 also introduced Manifold-managed spatial indices that allow spatial DBMS storage of drawings, images and surfaces in almost any DBMS, even those without native spatial capabilities. Release 8 also introduced the Manifold Spatial Extender for SQL Server, a server-side Manifold facility that enables SQL Server 2005 or later SQL Server editions to provide high performance spatial indices at no additional charge.

Release 8 greatly extended Manifold performance with much faster rendering of very large images, surfaces and progressive rendering for drawings. Massively parallel multiprocessor support was introduced with built in support for NVIDIA CUDA, enabling use of up to 512 stream processors for supercomputer performance, over a teraflop, in certain surface computations. Release 8 also introduced a new peak in Manifold engineering throughput, with simultaneous development on multiple code bases at once. Toward the end of Release 8 development, maintenance on the final update for Release 7x proceeded even as Release 8 was being brought to completion and teams began work on the next generation product as well.

Manifold System now runs in almost every country in the world and would not have been possible but for the international reach afforded by Internet and modern technology. In fact, this topic was originally written on a Dell laptop on a train running from Brussels to Amsterdam. It was teleported into the main documentation set via Internet. Internet has made it possible for friends from all over the world to send suggestions accompanied by detailed examples and wish lists of improvements desired in every new release.

User meetings throughout the world and interviews with numerous users representing North and South America, Europe, Africa, Asia and Pacific regions helped incorporate priorities from many elite users who combine many years' experience in GIS with a deep understanding of Manifold System. Manifold has been born a true citizen of the world, combining the best ideas from the finest GIS users in every continent. Manifold's corporate infrastructure and facilities have also always been deeply internationalized. To serve users with the best possible GIS at the lowest possible price, Manifold has not only solicited the intelligence and suggestions of all the world's GIS users, it has also invested into the best of all the world's most development-friendly regions to provide the most efficient possible locations and framework for modern software development.

For all of the tremendous benefits to world civilization that have arisen from Silicon Valley, benefits to the world have increased as the original microprocessor technology invented in Silicon Valley has expanded into major software and hardware development centers worldwide. Manifold has participated in that trend: every line of the millions of lines of code that have comprised Manifold products has been written in Europe and Asia, with corporate facilities and infrastructure located in the Caribbean, Europe and Asia. Manifold's Hong Kong corporate infrastructure and facilities together with upcoming language localization in new products continue the commitment to create a true, world GIS with support for the world's largest population centers throughout the world.

Release 5.00 was begun on Intel Pentium III machines with development switching to Pentium IV and Athlon towards the end of development. The native development environment at manifold.net began with Windows NT 4.0 and switched to Windows 2000 early in the R5 campaign with final development switching to Windows XP in the last year of 5.00 work. Enterprise Edition and 5.50 were coded entirely in XP with adjustments and backports as necessary to support earlier Windows editions. Current hardware used for development includes multi-core AMD and Intel processors using the most recent published and beta editions of Microsoft Windows. Current development runs at least two major releases simultaneously with work on future releases proceeding even as existing releases are maintained.

Manifold System is written entirely in Microsoft Visual C++ within the Microsoft Visual Studio development environment. In January of 2002 Manifold development moved to Microsoft's Visual Studio .NET development environment to take advantage of the improved .NET compiler. In 2003 all new code was migrated to .NET with steady conversion of existing code to .NET as well. Release 6.00 and 6.50 were coded entirely in Visual Studio.NET using Windows XP and Server 2003 on a mix of Intel and AMD based machines, including 64-bit AMD machines. Release 7x continued the move to modern Microsoft standards, such as use of .NET, multi-core processors and introduced 64-bit computing. Release 8 moved to deeply multi-processor development environments using multi-core main processors and hundreds of accessory stream processors.

Manifold System is completely new software. It is not simply a revision of older systems or a pasting-together of controls from other vendors but represents completely new code. The multi-version executable that now comprises Personal, Professional, Enterprise, Universal, Database Administrator, Ultimate and License Server Editions includes well over 2 million lines of source code. Each new major build of Manifold typically involves writing, editing or reviewing over 750,000 lines of code, as major portions of existing code are re-written from the ground up or refreshed to make the latest advancements possible without hindrance from legacy code.

Except for standard Microsoft development facilities and an ECW format importer/exporter, there is no third party code in Manifold. It is all entirely our work. The ECW importer/exporter is a small bit of code provided by the
ERMapper organization and is included within Manifold to emphasize our support for ERMapper. In addition, some color palettes and combinations have been designed by Cynthia Brewer of the ColorBrewer project.

Hundreds of people have contributed to the programming, graph theory, general mathematics and computational geometry within Manifold System. Even given the immense content of prior releases, yet again Release 8 has been the largest team project ever assembled at manifold.net. Once again, we can say that 8 is easily the most sophisticated software product any of us has ever created. As is typical for every major Manifold release, during the last months of final assembly of each edition our lives have revolved around the product. It has been stressful for friends and family but in the end, the long march is always a tremendous kick as we are privileged to watch each release come to life.

The manifold.net team would like to thank our many friends around the world for buying our products and for being so generous with their suggestions. If you purchased prior releases it is your investment that made the current release possible. By purchasing the current release you are financing the extensive work required to make the next releases possible. We are grateful you have helped bring this new phase of GIS into existence by funding our work. We are also grateful to the many users of beta test versions who patiently downloaded dozens of test builds and contributed suggestions and bug reports to help evolve the product.

We give greatest thanks to our friends and families who have put up with our absences and total focus on Manifold. Things like Manifold System do not happen without intense dedication. We are grateful our friends and families support us in this magnificent obsession. We hope you enjoy the result. It is our honor to work on your behalf.
About manifold.net

Manifold® System editions and extensions to Manifold® System are distributed by manifold.net. Product configurations and pricing are subject to change without notice and may be different than the following descriptions incorporated into Help at the time of writing. See the web site at www.manifold.net for the latest information on new products.

Manifold® System Products

Manifold System appears in several main product and option configurations:

Manifold® System License Server - Manages operation of multiple instances of Manifold (all of the same edition) in large enterprise environments via floating licenses. This is not a runtime or interactive product for doing GIS: it is a version of Manifold that works exclusively as a license server.

Manifold® System Personal Edition - The super-cool GIS as documented by this Help file including all Professional Edition features except the Internet Map Server (IMS) and the File - Export - Web Page command used to create IMS sites.


Manifold® System Database Administrator Edition - Enterprise Edition also equipped with the Administrator Console to provide administrative and managerial capabilities essential when using Manifold together with enterprise-class database servers to provide centralized geospatial storage and concurrent, multi-user editing. The Batch Export tool for Oracle Spatial databases makes loading large databases with drawings, images and surfaces a fast and simple process. Every significant installation using databases to provide centralized storage and multi-user editing should have at least one license of Database Administrator Edition that is used to set up the database for greater convenience and capability when used with other Manifold editions by other users.


Manifold® Spatial Extender for SQL Server - A free product that installs into Microsoft SQL Server installations to provide high performance spatial indices for spatial DBMS capability. For SQL Server 2005 (including SQL Server 2005 Express SP2) and newer SQL Server versions. Works automatically with Manifold® System to provide high performance, enterprise class spatial DBMS facilities.

Manifold® Business Tools - An optional extension to Manifold System that provides additional commands for working with drawings. The Business Tools extension adds functionality in several areas:

- Districts (Advanced) - Redistricting using a condensed, expert level dialog.
- Districts (Visual) - Redistricting via an easy-to-use visual, interactive dialog.
- Drive-Time Zones - Computation of drive-time zones via an easy-to-use, interactive dialog.
- Optimal Route - Finding an optimal route to designated locations using a condensed, expert level dialog.
- Optimal Route (Visual) - Finding an optimal route to designated locations via an easy-to-use, interactive dialog.
- Send Email - Automatically send email to map objects. Used to send spatially-targeted email.
- Topology Factory - An interactive dialog to view and repair common topological errors in drawings.
- Expanded programmatic access to routing and other functions.

If you do not have the optional Business Tools package enabled you will not have the above capabilities enabled within Manifold System.
**Manifold® Geocoding Tools** - An optional extension to Manifold System that provides street address geocoding capability as well as access to the Manifold Geocoding Database data provided on the Manifold downloads site for both US street address geocoding as well as the easy creation of drawings showing US streets. The **Geocoding Tools** package adds functionality in several areas:

- Street address geocoding within the United States using the Manifold Geocoding Database provided on the Manifold downloads site.
- Street address geocoding within North America or Europe using the appropriate edition of Microsoft's MapPoint product as a data source.
- Street address geocoding anywhere in the world using user-provided geocoding data extensions in either range or points of interest formats.
- Street address geocoding using web-based Manifold geocoding servers to provide geocoding services.
- Easy import of drawings showing US streets from the Manifold Geocoding Database provided on the Manifold downloads site as set forth in the Import Drawing - Geocoding Database topic.
- Geocoding Extensions may be used within SQL to allow use of street address geocoding functions within SQL queries.
- Programming objects such as the Geocoder object may be used from scripts or other programs or from within IMS applications.
- Street addresses and ZIP codes may be used in the Edit - Go To dialog.

If you do not have the optional **Geocoding Tools** package enabled you will not have the above capabilities enabled within Manifold System. Note that the **Geocoding Tools** package requires access to a geocoding data source appropriate to the geographic region of interest.

**Manifold® Surface Tools** - An optional extension to Manifold System that provides additional commands for working with surfaces. The **Surface Tools** extension adds functionality in several areas:

- A new Surface - Transform command dialog allows arbitrary transformation of surfaces, including computations that involve multiple surfaces such as subtracting one surface from another. A very rich collection of operators allows many different types of tasks to be accomplished.
- A Transfer Heights command enables rapid transfer of surface values from surfaces to points.
- The Visible Area command shows areas that are visible from given points. The height of viewing points may be automatically adjusted from a height field.
- New options in surface generation allow creating surfaces from drawings and tables using triangulation and Median-Polish Kriging.
- Expanded model choices for Kriging interpolation include Linear, Power, Rational and automatic model choices.
- New transform toolbar operators for interpolation of missing pixels, *Interpolate, Interpolate (Parameter)* and *Interpolate Row*.
- The ability to work with new Profiles and Elevations components that show cross-sectional cuts through a given path over a surface.
- Sophisticated watershed capabilities and other hydrology functions.
- Expanded programmatic access to surface operations.

If you do not have the optional **Surface Tools** package enabled you will not have the above capabilities enabled within Manifold System.

All Manifold System editions, the **Business Tools, Geocoding Tools** and **Surface Tools** packages are included within one installation package. Which product configuration launches when you start Manifold depends on the type of serial number that was provided when you installed Manifold and first launched the product. If you provide a Professional Edition serial number then the product will launch in Professional Edition mode. If you provide an Enterprise Edition serial number then the product will launch in Enterprise Edition mode.

This makes it easy to upgrade your installation. If you are a Professional Licensee and would like to upgrade to Enterprise Edition, visit the Manifold Online Store to order an upgrade product. You will receive a new serial number that you can use to upgrade your installation without the need to download a new installation.

**Runtime Licenses**
Manifold® System may be licensed in runtime mode by acquiring a runtime license of either Professional or Universal editions. Runtime licenses are not available for Personal, Enterprise, or Database Administrator editions.

A runtime license allows Manifold System to be used as an Internet Map Server (IMS) or via the applications program interface (API), but not using the interactive graphical user interface (GUI). Runtime licenses are intended to provide reduced-cost licenses for IMS applications or for third party programming where Manifold is embedded within a third party application.

**Manifold® System Professional Runtime** - Provides all IMS and API capabilities of Professional Edition without any interactive GUI capabilities.

**Manifold® System Universal Runtime** - Provides all IMS and API capabilities of Universal Edition without any interactive GUI capabilities.

Visit the [www.manifold.net](http://www.manifold.net) website for the latest news on Manifold products for data visualization, data discovery, business intelligence, mapping and more. Manifold tools work with Microsoft operating systems and other Microsoft products to deliver a seamless enhancement of your ability to see, manipulate, and analyze data using a visual, point-and-click interface.
This topic contains miscellaneous legal notices, some of which may be applicable only if you are a source code licensee of Manifold System.

As noted in the About Manifold System topic, this product is entirely our work with the exception of a few items, for which credit is due:

**ColorBrewer**

Your Manifold distribution may contain color palettes (choices of colors) that originate in the ColorBrewer project. If so, the following license terms and provisions apply:

Apache-Style Software License for ColorBrewer Color Schemes
Version 1.1

Copyright (c) 2002 Cynthia Brewer, Mark Harrower, and The Pennsylvania State University. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions as source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. The end-user documentation included with the redistribution, if any, must include the following acknowledgment:

   "This product includes color specifications and designs developed by Cynthia Brewer (http://colorbrewer.org/)."

Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.

4. The name "ColorBrewer" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact Cynthia Brewer at cbrewer@psu.edu.

5. Products derived from this software may not be called "ColorBrewer", nor may "ColorBrewer" appear in their name, without prior written permission of Cynthia Brewer.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL CYNTHIA BREWER, MARK HARROWER, OR THE PENNSYLVANIA STATE UNIVERSITY BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Index

..map 59
..mid/.mif 1742, 1825
2 17
2008 2262, 2300
3 3
32-bit and 64-bit Manifold Editions 9
A A Flashy Demo - Web Queries and KML 3835
About Geocoding 203
About Manifold System 4341
About manifold.net 4344
About Networks 2068
About the Sample Images 4175
Acres 4120
Action Queries 1377
Activate Extension 3446
Activation Keys 4227
Activation Keys and Serial Numbers 3443, 4227
Active Columns 1300
Active Columns using JScript 4117
Active Columns using VBScript 4113
active layer tab 89
active window 37
ActiveX Control 2505
Add Margin 975
Add Points with Instant Data 3652
Add Rank Column Dialog 1447
Adding a New Layer 89
Adding Labels to a Map Manually 135
Adding Legends 138
Adding or Deleting Fields in Tables 1331
Adding Points, Lines and Areas 524
Adding Shapes 530
Adding Text Labels 135
Add-Ins 2167
address geocoding 1407
Administrator Console 3282
ADG.NET 1226
ADRG 1796
Advanced 3103
Aggregate Functions 3569
AI1836 1407
Ajax 4296
Albers Conical Equal Area 2013
Align Items in Layouts 3931
Aligning Labels 1124
Alignment Toolbar 3457
ALL 3591
ALL DISTINCT SKIP TOP Quantifiers 3591
all visible layers 89
ALL, ANY, SOME Quantifiers 3589
alpha channel 859
alpha transparency 859
ALT- Tab 37
ALTER TABLE Statement 3570
Altitude 1470
Altitude and Bathymetry 1470
Altitude, Aeronautical 1470
Analysis 17, 1565
Analyzer 2590
AnalyzerValue 2602
AnalyzerValueSet 2603
AnalyzerValueSetColumn 2604
Animation 2452
ANSI SQL 3561
ANY 3589
Append 1335
Appendices 4279
Application 2605
ArcGIS 318, 2223
ArcGrid 1789
ArcINFO DEM 1790
Arcs 4315
ArcSDE 312, 318, 2223
Area Formatting 444
Area of a Parcel 4120
Area Overlays 1497
Area Styles and Size 449
Areas and Boundary Lines 451
aspect 1541
Assign Projection 3075
Attach to 610
Attach to Self 610
Attaching External Tables to Drawings 1272
AutoCAD .dwg 1707
AutoCAD .dx 1708
Autocomplete with ALT 558
Autocompletion in Queries 1363
Autorefresh View 3155
autoscroll 37
AUX 226, 3075, 3080
Azimuthal 1980
Azimuthal Equidistant 1991
B BETWEEN AND Operator 3593
Beware of Counterfeit Software 4313
Bezier 701
Bipolar Oblique Conformal Conic 2015
blob 312
4349
<table>
<thead>
<tr>
<th>Term</th>
<th>Page</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMP</td>
<td>1832</td>
<td></td>
</tr>
<tr>
<td>BNA</td>
<td>1700</td>
<td></td>
</tr>
<tr>
<td>Bonne</td>
<td>2017</td>
<td></td>
</tr>
<tr>
<td>Bookmarks</td>
<td>3105</td>
<td></td>
</tr>
<tr>
<td>Border</td>
<td>897, 2521</td>
<td></td>
</tr>
<tr>
<td>Border Buffers</td>
<td>615</td>
<td>color well</td>
</tr>
<tr>
<td>Boundaries</td>
<td>616</td>
<td>color wheel</td>
</tr>
<tr>
<td>Boundary Lines</td>
<td>451</td>
<td>ColorBrewer</td>
</tr>
<tr>
<td>Bounded Areas</td>
<td>617</td>
<td>Colorize</td>
</tr>
<tr>
<td>Bounding Boxes</td>
<td>620</td>
<td>Colors as Hue, Saturation and Brightness</td>
</tr>
<tr>
<td>Branch</td>
<td>2611</td>
<td>ColorSet</td>
</tr>
<tr>
<td>BranchSet</td>
<td>2613</td>
<td>Column</td>
</tr>
<tr>
<td>Breakpoints</td>
<td>3107</td>
<td>ColumnAlign</td>
</tr>
<tr>
<td>Brightness</td>
<td>836</td>
<td>ColumnCategory</td>
</tr>
<tr>
<td>Brightness / Contrast</td>
<td>3319</td>
<td>ColumnFlag</td>
</tr>
<tr>
<td>Buffers</td>
<td>621</td>
<td>ColumnFormat</td>
</tr>
<tr>
<td>BufferType</td>
<td>2522</td>
<td>ColumnFormatNeg</td>
</tr>
<tr>
<td>Business Tools</td>
<td>712</td>
<td>ColumnFormatPos</td>
</tr>
<tr>
<td>ByteOrder</td>
<td>2523</td>
<td>ColumnLookupSet</td>
</tr>
<tr>
<td>C#</td>
<td>2380</td>
<td>Columns</td>
</tr>
<tr>
<td>Cached and Uncached Components</td>
<td>2193</td>
<td>ColumnSet</td>
</tr>
<tr>
<td>Calculations in Queries</td>
<td>1375</td>
<td>ColumnType</td>
</tr>
<tr>
<td>CameraType</td>
<td>2524</td>
<td>Combine a Surface and a Drawing in a Map</td>
</tr>
<tr>
<td>Cascade</td>
<td>37</td>
<td>Combo Box</td>
</tr>
<tr>
<td>CASE</td>
<td>3594</td>
<td>combo boxes</td>
</tr>
<tr>
<td>CASE Operator</td>
<td>3594</td>
<td>ComboControl</td>
</tr>
<tr>
<td>Cassini</td>
<td>2024</td>
<td>Command Button</td>
</tr>
<tr>
<td>CAST</td>
<td>3596</td>
<td>Command Filters</td>
</tr>
<tr>
<td>CAST Operator</td>
<td>3596</td>
<td>Command Line Options</td>
</tr>
<tr>
<td>Center Point</td>
<td>3475</td>
<td>CommandControl</td>
</tr>
<tr>
<td>Centroids</td>
<td>624</td>
<td>Comments</td>
</tr>
<tr>
<td>Change Projection</td>
<td>3080</td>
<td>Common Dialog Control</td>
</tr>
<tr>
<td>Changing a Component's Projection</td>
<td>1923</td>
<td>Community</td>
</tr>
<tr>
<td>Changing Field Types in Tables</td>
<td>1332</td>
<td>Comparison with ANSI SQL</td>
</tr>
<tr>
<td>Changing Palettes</td>
<td>865</td>
<td>Comparison with Jet SQL</td>
</tr>
<tr>
<td>ChannelInterleaving</td>
<td>2525</td>
<td>Compile to DLL</td>
</tr>
<tr>
<td>Channels</td>
<td>830, 932</td>
<td>Component</td>
</tr>
<tr>
<td>Chart</td>
<td>2615</td>
<td>Component Windows</td>
</tr>
<tr>
<td>Charting</td>
<td>17</td>
<td>ComponentControl</td>
</tr>
<tr>
<td>Charts</td>
<td>1453</td>
<td>components</td>
</tr>
<tr>
<td>Check Box</td>
<td>2421</td>
<td>Components</td>
</tr>
<tr>
<td>Check for Updates</td>
<td>3449</td>
<td>ComponentSet</td>
</tr>
<tr>
<td>CheckControl</td>
<td>3034</td>
<td>ComponentType</td>
</tr>
<tr>
<td>Click Events</td>
<td>2388</td>
<td>Composing Complex Images in Layers</td>
</tr>
<tr>
<td>Clip</td>
<td>2001</td>
<td>Compressed Images</td>
</tr>
<tr>
<td>Clip coordinates</td>
<td>2001</td>
<td>Compression</td>
</tr>
<tr>
<td>Clip with (Intersect)</td>
<td>646</td>
<td>ComputationMode</td>
</tr>
<tr>
<td>Clip with (Subtract)</td>
<td>646</td>
<td>conforal</td>
</tr>
<tr>
<td>Clipboard</td>
<td>37</td>
<td>Conic</td>
</tr>
<tr>
<td>Close</td>
<td>37</td>
<td>Connect Clusters</td>
</tr>
<tr>
<td>Close All</td>
<td>37</td>
<td>Constrained Triangulation</td>
</tr>
<tr>
<td>Clusters</td>
<td>649</td>
<td>Contacting manifold.net</td>
</tr>
<tr>
<td>Context Menus</td>
<td>ControlStyle</td>
<td></td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>----------------------------------------</td>
<td></td>
</tr>
<tr>
<td>context sensitive</td>
<td>ControlStyleCombo</td>
<td></td>
</tr>
<tr>
<td>Contours</td>
<td>ControlStyleList</td>
<td></td>
</tr>
<tr>
<td>Contract</td>
<td>ControlTrackEventArgs</td>
<td></td>
</tr>
<tr>
<td>Control - Animation</td>
<td>Convert To</td>
<td></td>
</tr>
<tr>
<td>Control - Chart</td>
<td>ConverterItemSet</td>
<td></td>
</tr>
<tr>
<td>Control - Check Box</td>
<td>ConverterProperty</td>
<td></td>
</tr>
<tr>
<td>Control - Combo Box</td>
<td>ConverterPropertySet</td>
<td></td>
</tr>
<tr>
<td>Control - Command Button</td>
<td>ConvertPolicy</td>
<td></td>
</tr>
<tr>
<td>Control - Common Dialog Control</td>
<td>ConvertPrompt</td>
<td></td>
</tr>
<tr>
<td>Control - Cool Bar</td>
<td>Convex Hull</td>
<td></td>
</tr>
<tr>
<td>Control - Date / Time Picker</td>
<td>Cookie Cutter a Large Image with Transfer Selection</td>
<td></td>
</tr>
<tr>
<td>Control - Frame</td>
<td>Cool Bar</td>
<td></td>
</tr>
<tr>
<td>Control - Horiz / Vert Scroll Bars</td>
<td>CoordinateBand</td>
<td></td>
</tr>
<tr>
<td>Control - Image Combo Box</td>
<td>CoordinateConverter</td>
<td></td>
</tr>
<tr>
<td>Control - Image List</td>
<td>Coordinates</td>
<td></td>
</tr>
<tr>
<td>Control - Insert ActiveX Control</td>
<td>Coordinates in Projected Maps</td>
<td></td>
</tr>
<tr>
<td>Control - List Box</td>
<td>Coordinates Tutorial</td>
<td></td>
</tr>
<tr>
<td>Control - List View</td>
<td>CoordinateSystem</td>
<td></td>
</tr>
<tr>
<td>Control - Masked Text Box</td>
<td>CoordinateSystemParameter</td>
<td></td>
</tr>
<tr>
<td>Control - Month View</td>
<td>CoordinateSystemParameterSet</td>
<td></td>
</tr>
<tr>
<td>Control - Multimedia Control</td>
<td>CoordinateSystemParameterType</td>
<td></td>
</tr>
<tr>
<td>Control - Option Button</td>
<td>CoordinateSystemSet</td>
<td></td>
</tr>
<tr>
<td>Control - Picture Clip</td>
<td>Copy</td>
<td></td>
</tr>
<tr>
<td>Control - Progress Bar</td>
<td>Copy and Paste</td>
<td></td>
</tr>
<tr>
<td>Control - Rich Text Box</td>
<td>Copy and Paste As</td>
<td></td>
</tr>
<tr>
<td>Control - Slider</td>
<td>Copy and Paste in Images</td>
<td></td>
</tr>
<tr>
<td>Control - Static Text Box</td>
<td>Create a Circular Feathered Image</td>
<td></td>
</tr>
<tr>
<td>Control - Status Bar</td>
<td>Create a Geocoded Table from a Map</td>
<td></td>
</tr>
<tr>
<td>Control - System Info Control</td>
<td>Create a Geocoded Table</td>
<td></td>
</tr>
<tr>
<td>Control - Tab Strip</td>
<td>Create a Linked Drawing from a Geocoded Table3792</td>
<td></td>
</tr>
<tr>
<td>Control - Text Box</td>
<td>Create a Map from a Geocoded Table</td>
<td></td>
</tr>
<tr>
<td>Control - Tool Bar</td>
<td>Create a Print Layout from a Table</td>
<td></td>
</tr>
<tr>
<td>Control - Tools (Advanced) Toolbar</td>
<td>Create a Projected US Map</td>
<td></td>
</tr>
<tr>
<td>Control - Tree Toolbar</td>
<td>Create a Table and Add Records</td>
<td></td>
</tr>
<tr>
<td>Control - UpDown</td>
<td>CREATE DRAWING Statement</td>
<td></td>
</tr>
<tr>
<td>Control Points</td>
<td>Create Index Drawing</td>
<td></td>
</tr>
<tr>
<td>ControlAlignment</td>
<td>CREATE TABLE Statement</td>
<td></td>
</tr>
<tr>
<td>ControlAlignmentText</td>
<td>CREATE VIEW</td>
<td></td>
</tr>
<tr>
<td>ControlAppearance</td>
<td>CREATE VIEW Statement</td>
<td></td>
</tr>
<tr>
<td>ControlBorderStyle</td>
<td>Creating a New Map</td>
<td></td>
</tr>
<tr>
<td>ControlCheckValue</td>
<td>Creating a New Project</td>
<td></td>
</tr>
<tr>
<td>ControlMouseButton</td>
<td>Creating a Project</td>
<td></td>
</tr>
<tr>
<td>ControlMouseEventArgs</td>
<td>Creating a Web Site</td>
<td></td>
</tr>
<tr>
<td>ControlMouseMode</td>
<td>Creating an Enterprise Server</td>
<td></td>
</tr>
<tr>
<td>ControlMousePointer</td>
<td>Creating Bordered Lines</td>
<td></td>
</tr>
<tr>
<td>ControlMultiSelect</td>
<td>Creating Drawings from Geocoded Tables</td>
<td></td>
</tr>
<tr>
<td>ControlOleDrag</td>
<td>Creating Labels from Fields</td>
<td></td>
</tr>
<tr>
<td>ControlOleDrop</td>
<td>Creating New Components</td>
<td></td>
</tr>
<tr>
<td>ControlPoint</td>
<td>Creating New Tables</td>
<td></td>
</tr>
<tr>
<td>ControlPointSet</td>
<td>Creating Surfaces from Drawings and Tables</td>
<td></td>
</tr>
<tr>
<td>ControlScrollBars</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Index

Drawings - Context Menus 3505 Editing a Surface for Visual Effect 3945
Drawings - Object Coordinates 3508 Editing Data in Tables 1311
Drawings - Object Fields 3510 Editing Drawings 517
Drawings and Databases 162 Editing Forms 2403
Drawings and Tables 168 Editing Images 906
Drawings, Images and Maps 86 Editing Intrinsic Fields in Tables 1321
DrawingSelectionSet 2668 Editing Objects 540
DrawingWindow 2669 Editing Palettes 947
Drive-Time Zones 741 Editing Surfaces 1506
DROP DRAWING Statement 3577 Editing with Snap 552
DROP TABLE 3578 Elevation 2677
DROP TABLE Statement 3578 Ellipsoid 1968, 2680
DROP VIEW 3579 EllipsoidSet 2681
DROP VIEW Statement 3579 EMF 1832, 1836
DSSAtom 2672 Enclosing Circles 635
DSSAtomSet 2674 Enclosing Rectangles 636
DSSAtomType 2542 Enterprise Edition 2175
DSSHedge 2543 EOS 1793
DSSJunction 2544 equal-area 1975
DSSQuery 2675 Equalize 3337
DSSQueryEntry 2676 equidistant 1975
DTED 1792 Equidistant Conic 2018
DWG 1707 ERDAS 226, 3075, 3080

E
Eckert IV 2038 Errors 3173
Eckert VI 2040 ESC 37
ECW 1832, 1838 ESRI geodatabase 1727
Edit - Advanced 3103 ESRI Geodatabases 318
Edit - Assign Projection 3075 Essays 4281
Edit - Bookmarks 3105 ETAK 1738
Edit - Breakpoints 3107 ETAK MapBase 1738
Edit - Change Projection 3080 Euler 2068
Edit - Cut / Copy 3090 EventArgs 2682
Edit - Delete / Delete All 3092 Example
Edit - Find / Find Next 3101 Example Selection Tools 101
Edit - Go / Go To 3109 Example Configuring SQL Server 2008 2268
Edit - Instant Data 3125 Example Linking a Drawing from SQL Server 20082282
Edit - Modify Selection 3098 Example Storing a Drawing in Manifold Spatial DBMS 2344
Edit - Paste / Paste Append 3091 Example Storing a Drawing in Oracle 2239
Edit - Replace 3102 Example Storing a Drawing in SQL Server 20082278
Edit - Save, Load Mask/Channel 3099 Example Storing a Surface in Manifold Spatial DBMS 2364
Edit - Select All / None / Inverse 3093 Example Storing an Image in Manifold Spatial DBMS 2355
Edit - Select by Type 3097 Example Storing an Image in Oracle 2253
Edit - Select Mode 3095 Example Storing an Image in SQL Server 20082331
Edit - Select Objects 3096 Example Tracing Virtual Earth into SQL Server 2008 2300
Edit - Shared Edit 3118 Examples 3633
Edit - Snap To 3115, 3111 Excel 1801
Edit - Template 3127 EXCEPT 3598
Edit - Undo / Redo 3089 EXCEPT Operator 3598
Edit a Table with the Transform Toolbar 3765 EXIF 1774
Edit Active Column 3386
Index

Finding Data in Tables 1306
Flip Horizontally 985
Flip Vertically 985
Fluoresce 3344
focus 37
Folder 2734
Font 2736
For Experienced GIS Users 296
Form 2737
Form Controls 2399
Form Properties 2397
Format 2739
Format Toolbar 3470
Formatting 132
Formatting Columns 1257
Formatting Drawings 432
Formatting Individual Labels 1112
Formatting Labels 1105
Formatting Lines in a Linked Drawing 3804
FormatType 2545
FormatValue 2742
FormatValueSet 2743
Forms 2395
Forms and Scripts 2413
FoxPro 4196
Frame 2433
FrameControl 3048
Freehand Tracing 1047
FROM Clause 3600
Full Screen 37, 3153
G
Gabriel Network 660
Gall 2028
Gamma 3347
Gauss Kruger 1983
Gaussian Blur 3348
GDF 1716
General Projections Concepts 1964
geocentric 1968
Geocode 3388
GeocodeLevel 2546
GeocodeMatch 2744
GeocodeMatchSet 2745
Geocoder Object 2746
GeocodeStatus 2547
Geocoding 1403, 1436
Geocoding Data Extensions 1428
Geocoding Data Sources 1421
Geocoding Extensions 3551
Geocoding Server Interface 2512
Geocoding Tools 1405
Geocoding with MapPoint 1424
Geodatabase 318, 1727, 2223
Geodatabases 318
Geom 2747
Geometry in Tables 1275
Georeferencing 4320
Georegister a Scanned Paper Map 3661
Georegistration an Image to a Drawing 1944
Georegistration an Image to Known Coordinates 1945
GeoTIFF 1832, 1841
Go / Go To 3109
Golden Software 226, 3075, 3080
Goode’s Homolosine 2041
Google 872
Google Earth 3701
GPS 250
GPS Console 3176
GPS Receivers 250
Grabber 3475
Gradient Tool 963
Graph theory 2068
Graphics Projections 1983
Graticule 3137
Graticule Labels in a Print Layout 3926
Grayscale 986
Grayscale Images 789
Grid 3141
Griddascii 1784
Grids 4318
GROUP BY Clause 3602
GS50 Projection 1988
H
Hammer 2046
HDF EOS 1793
Help - About 3448
Help - Activate 3443
Help - Activate Extension 3446
Help - Check for Updates 3449
Help - Contents 3450
Help - Index 3451
Help - Manifold on the Web 3453
HABING Clause 3603
HAVING Clause 3603
<table>
<thead>
<tr>
<th>Help - Search</th>
<th>3452</th>
<th>Images</th>
<th>17, 775</th>
</tr>
</thead>
<tbody>
<tr>
<td>Help Links</td>
<td>4314</td>
<td>Images and Channels</td>
<td>830</td>
</tr>
<tr>
<td>High Pass</td>
<td>987</td>
<td>Images and Surfaces</td>
<td>872</td>
</tr>
<tr>
<td>Highway Shield Labels</td>
<td>1134</td>
<td>Images can be Inefficient</td>
<td>4288</td>
</tr>
<tr>
<td>History</td>
<td>2753</td>
<td>ImageSelectionSet</td>
<td>2759</td>
</tr>
<tr>
<td>Horizontal Scroll bar</td>
<td>2448</td>
<td>ImageType</td>
<td>2549</td>
</tr>
<tr>
<td>hot scroll</td>
<td>37</td>
<td>IMAGINE</td>
<td>1784</td>
</tr>
<tr>
<td>How to Install this Documentation</td>
<td>3</td>
<td>Import</td>
<td>17, 2760, 3068</td>
</tr>
<tr>
<td>How to Print</td>
<td>236</td>
<td>Import a Drawing from the Geocoding Database</td>
<td>3693</td>
</tr>
<tr>
<td>HScrollControl</td>
<td>3050</td>
<td>Import a Projected Shapefile</td>
<td>4015</td>
</tr>
<tr>
<td>Hue</td>
<td>836</td>
<td>Import a Raw Binary File - NLCD</td>
<td>4022</td>
</tr>
<tr>
<td>Hue / Saturation</td>
<td>3332</td>
<td>Import a Shapefile</td>
<td>4006</td>
</tr>
<tr>
<td>Hyperlinks</td>
<td>429</td>
<td>Import a VMAP Level 1 File</td>
<td>4029</td>
</tr>
<tr>
<td>I</td>
<td></td>
<td>Import and Export</td>
<td>1679</td>
</tr>
<tr>
<td>IBM DB2</td>
<td>312</td>
<td>Import Component</td>
<td>1691</td>
</tr>
<tr>
<td>IBM DB2 Express-C Edition</td>
<td>353</td>
<td>Import Drawing - ADF</td>
<td>1698</td>
</tr>
<tr>
<td>Identity</td>
<td>593</td>
<td>Import Drawing - ADO .NET / ODBC / OLE DB</td>
<td>1699</td>
</tr>
<tr>
<td>If the Selection Pattern Interferes</td>
<td>101</td>
<td>Import Drawing - BNA</td>
<td>1700</td>
</tr>
<tr>
<td>Image</td>
<td>2754</td>
<td>Import Drawing - CSV</td>
<td>1701</td>
</tr>
<tr>
<td>Image - Brightness / Contrast</td>
<td>3319</td>
<td>Import Drawing - DB</td>
<td>1702</td>
</tr>
<tr>
<td>Image - Color Balance</td>
<td>3322</td>
<td>Import Drawing - DBF</td>
<td>1703</td>
</tr>
<tr>
<td>Image - Colorize</td>
<td>3320</td>
<td>Import Drawing - DGN</td>
<td>1704</td>
</tr>
<tr>
<td>Image - Convert To</td>
<td>3384</td>
<td>Import Drawing - DSG</td>
<td>1705</td>
</tr>
<tr>
<td>Image - Create Index Drawing</td>
<td>3323</td>
<td>Import Drawing - DSN</td>
<td>1706</td>
</tr>
<tr>
<td>Image - Diffuse</td>
<td>3330</td>
<td>Import Drawing - DWG</td>
<td>1707</td>
</tr>
<tr>
<td>Image - Dither</td>
<td>3334</td>
<td>Import Drawing - DXF</td>
<td>1708</td>
</tr>
<tr>
<td>Image - Download</td>
<td>3335</td>
<td>Import Drawing - E00</td>
<td>1715</td>
</tr>
<tr>
<td>Image - Equalize</td>
<td>3337</td>
<td>Import Drawing - ETAK MapBase</td>
<td>1738</td>
</tr>
<tr>
<td>Image - Filter</td>
<td>3339</td>
<td>Import Drawing - GDF</td>
<td>1716</td>
</tr>
<tr>
<td>Image - Fluoresce</td>
<td>3344</td>
<td>Import Drawing - Geocoding Database</td>
<td>1717</td>
</tr>
<tr>
<td>Image - Gamma</td>
<td>3347</td>
<td>Import Drawing - Geodatabase</td>
<td>1727</td>
</tr>
<tr>
<td>Image - Gaussian Blur</td>
<td>3348</td>
<td>Import Drawing - GML</td>
<td>1728</td>
</tr>
<tr>
<td>Image - Hue / Saturation</td>
<td>3332</td>
<td>Import Drawing - HTML</td>
<td>1739</td>
</tr>
<tr>
<td>Image - Motion Blur</td>
<td>3355</td>
<td>Import Drawing - KML KMZ</td>
<td>1734</td>
</tr>
<tr>
<td>Image - Noise</td>
<td>3346</td>
<td>Import Drawing - LULC</td>
<td>1735</td>
</tr>
<tr>
<td>Image - Posterize</td>
<td>3358</td>
<td>Import Drawing - MDB</td>
<td>1740</td>
</tr>
<tr>
<td>Image - Quantize</td>
<td>3359</td>
<td>Import Drawing - MFD</td>
<td>1741</td>
</tr>
<tr>
<td>Image - Relief</td>
<td>3362</td>
<td>Import Drawing - MIF</td>
<td>1742</td>
</tr>
<tr>
<td>Image - Relink / Unlink</td>
<td>3364</td>
<td>Import Drawing - MWS</td>
<td>1743</td>
</tr>
<tr>
<td>Image - Resize</td>
<td>3383</td>
<td>Import Drawing - NTAD</td>
<td>1744</td>
</tr>
<tr>
<td>Image - Simplify</td>
<td>3366</td>
<td>Import Drawing - NTF</td>
<td>1745</td>
</tr>
<tr>
<td>Image - Threshold</td>
<td>3368</td>
<td>Import Drawing - SDTS</td>
<td>1746</td>
</tr>
<tr>
<td>Image - Threshold Color</td>
<td>3375</td>
<td>Import Drawing - SHP, Shapefiles</td>
<td>1748</td>
</tr>
<tr>
<td>Image - Tile</td>
<td>3381</td>
<td>Import Drawing - TAB</td>
<td>1755</td>
</tr>
<tr>
<td>Image Combo Box</td>
<td>2456</td>
<td>Import Drawing - TAI F</td>
<td>1757</td>
</tr>
<tr>
<td>Image Editing Tools</td>
<td>948</td>
<td>Import Drawing - TIGER/Line</td>
<td>1758</td>
</tr>
<tr>
<td>Image Effects in Maps</td>
<td>3982</td>
<td>Import Drawing - UDL</td>
<td>1760</td>
</tr>
<tr>
<td>Image Libraries</td>
<td>809</td>
<td>Import Drawing - VCT</td>
<td>1761</td>
</tr>
<tr>
<td>Image List</td>
<td>2454</td>
<td>Import Drawing - VMAP</td>
<td>1762</td>
</tr>
<tr>
<td>Image Menu</td>
<td>3317</td>
<td>Import Drawing - Wkx</td>
<td>1766</td>
</tr>
<tr>
<td>Image Server Interface</td>
<td>2515</td>
<td>Import Drawing - XLS</td>
<td>1767</td>
</tr>
<tr>
<td>Image Types</td>
<td>789</td>
<td>Import Image - JPG JPEG</td>
<td>1774</td>
</tr>
<tr>
<td>Image Window Menus and Controls</td>
<td>783</td>
<td>Import Image - SID MrSID</td>
<td>1776</td>
</tr>
</tbody>
</table>
Import Image - TIF [Various types] 1781
Import Landsat Files and Create Composite RGB Image 4073
Import Surface - ADF 1789
Import Surface - DTED 1792
Import Surface - HDF EOS 1793
Import Surface - Northwood GRC / GRD 1794
Import Surface - Raw Binary 1796
Import Surface - Raw Text 1795
Import Surface - SRTM 1798
Import Surface - XYZ 1801
Import Table - CSV 1807
ImportAdf 2761
ImportAdoNet 2763
ImportAdoNetOdbc 2764
ImportAdoNetOleDb 2765
ImportAdoNetOracle 2766
ImportAdoNetSqlServer 2768
ImportAdrg 2770
ImportAvhrr 2771
ImportBil 2772
ImportBmp 2773
ImportBna 2774
ImportCadrgCib 2775
ImportCeosSeaWifs 2776
ImportCsv 2777
ImportDb 2779
ImportDb2 2781
ImportDbf 2783
ImportDem 2784
ImportDemGlobe 2785
ImportDemGtopo30 2786
ImportDgn 2787
ImportDig 2788
ImportDoq 2789
ImportDsn 2790
ImportDted 2791
ImportDwg 2792
ImportDxf 2794
ImportE00 2796
ImportEcw 2798
ImportEmf 2799
ImportEnvi 2800
ImportEras 2801
ImportErdas 2802
ImportErdasImagine 2803
ImportErs 2804
ImportGdb 2805
ImportGif 2807
ImportGeoSpot 2809
ImportGml 2810
ImportGrass 2812
ImportGrd 2813
ImportGrdSurfer 2814
ImportGxf 2815
ImportHdf 2816
ImportHdfEos 2817
ImportHdfSeaWifs 2819
ImportHtml 2820
ImportImdisp 2821
Importing .dbf 1801
Importing and Linking Tables 1226
Importing Drawings 1691
Importing from Excel 1801
Importing Georegistered Images 1767
Importing Images 1767
Importing Surfaces 1784
Importing Tables 1801
ImportJpeg 2822
ImportKml 2823
ImportLas 2824
ImportLulc 2825
ImportLulcGiras 2826
ImportMapBase 2827
ImportMdb 2828
ImportMfd 2829
ImportMif 2830
ImportMvs 2831
ImportNetcdf 2832
ImportNef 2833
ImportNorthwood 2834
ImportNtad 2835
ImportNtif 2836
ImportOdbc 2837
ImportOleDb 2839
ImportOracle 2841
ImportPcx 2843
ImportPix 2844
ImportPng 2845
ImportPostgresql 2846
ImportPpm 2848
ImportRawAscii 2849
ImportRawBinary 2851
ImportRst 2853
ImportS57 2854
ImportSdts 2855
ImportSgl 2857
ImportShp 2858
ImportSid 2860
ImportSpot 2861
ImportSqlserver 2862
ImportStrem 2864
ImportSun 2865
ImportTab 2866
ImportTiff 2867
<table>
<thead>
<tr>
<th>Term</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ImportTga</td>
<td>2868</td>
</tr>
<tr>
<td>ImportTiff</td>
<td>2869</td>
</tr>
<tr>
<td>ImportTiger</td>
<td>2870</td>
</tr>
<tr>
<td>ImportTxt</td>
<td>2871</td>
</tr>
<tr>
<td>ImportUdl</td>
<td>2872</td>
</tr>
<tr>
<td>ImportVct</td>
<td>2873</td>
</tr>
<tr>
<td>ImportVmap</td>
<td>2874</td>
</tr>
<tr>
<td>ImportWk</td>
<td>2876</td>
</tr>
<tr>
<td>ImportXml</td>
<td>2877</td>
</tr>
<tr>
<td>ImportXyz</td>
<td>2878</td>
</tr>
<tr>
<td>IMS Config.txt Options</td>
<td>2103</td>
</tr>
<tr>
<td>IMS Queries</td>
<td>2107</td>
</tr>
<tr>
<td>IN Operator</td>
<td>3604</td>
</tr>
<tr>
<td>index drawing</td>
<td>809</td>
</tr>
<tr>
<td>Index Drawing</td>
<td>3323</td>
</tr>
<tr>
<td>Info</td>
<td>3181</td>
</tr>
<tr>
<td>Inner Buffers</td>
<td>641</td>
</tr>
<tr>
<td>INNER JOIN Operator</td>
<td>3605</td>
</tr>
<tr>
<td>Insert ActiveX Control</td>
<td>2505</td>
</tr>
<tr>
<td>INSERT INTO Statement</td>
<td>3580</td>
</tr>
<tr>
<td>Installation</td>
<td>5</td>
</tr>
<tr>
<td>Installing and Activating a Manifold Extension</td>
<td>4255</td>
</tr>
<tr>
<td>Installing Oracle</td>
<td>2235</td>
</tr>
<tr>
<td>Instant Data</td>
<td>536, 3125</td>
</tr>
<tr>
<td>Intermediate Levels and Pyramids</td>
<td>799</td>
</tr>
<tr>
<td>Interpolate</td>
<td>1523</td>
</tr>
<tr>
<td>Intersect</td>
<td>593</td>
</tr>
<tr>
<td>INTERSECT</td>
<td>3607</td>
</tr>
<tr>
<td>Intersect Lines</td>
<td>663</td>
</tr>
<tr>
<td>INTERSECT Operator</td>
<td>3607</td>
</tr>
<tr>
<td>Intersection Overlays</td>
<td>3874</td>
</tr>
<tr>
<td>Intersection Points</td>
<td>642</td>
</tr>
<tr>
<td>Intrinsic Fields in Tables</td>
<td>1244</td>
</tr>
<tr>
<td>Introduction</td>
<td>5, 17</td>
</tr>
<tr>
<td>Introduction to Decision Support</td>
<td>1444</td>
</tr>
<tr>
<td>Invert</td>
<td>3353</td>
</tr>
<tr>
<td>Invisible Pixels</td>
<td>838</td>
</tr>
<tr>
<td>Invisible Pixels and Selection</td>
<td>852</td>
</tr>
<tr>
<td>Invisible Pixels in Masks</td>
<td>838</td>
</tr>
<tr>
<td>Invisible Pixels Saved Selection</td>
<td>838</td>
</tr>
<tr>
<td>Invisible Pixels vs. RGBA Pixel Transparency</td>
<td>838</td>
</tr>
<tr>
<td>IronPython</td>
<td>2380</td>
</tr>
<tr>
<td>IS3609</td>
<td></td>
</tr>
<tr>
<td>IS NULL</td>
<td>3611</td>
</tr>
<tr>
<td>IS NULL Operator</td>
<td>3611</td>
</tr>
<tr>
<td>IS Operator</td>
<td>3609</td>
</tr>
<tr>
<td>J</td>
<td></td>
</tr>
<tr>
<td>John Parr Snyder</td>
<td>2060</td>
</tr>
<tr>
<td>Join Lines</td>
<td>640</td>
</tr>
<tr>
<td>JPEG</td>
<td>1832</td>
</tr>
<tr>
<td>JPEG 2000</td>
<td>1838</td>
</tr>
<tr>
<td>JpegCompression</td>
<td>2550</td>
</tr>
<tr>
<td>Key Ideas</td>
<td>32</td>
</tr>
<tr>
<td>Key Ideas in GIS</td>
<td>21</td>
</tr>
<tr>
<td>Key Windows and Dialogs</td>
<td>267</td>
</tr>
<tr>
<td>Keyboard</td>
<td>37</td>
</tr>
<tr>
<td>Keyboard Shortcuts</td>
<td>37</td>
</tr>
<tr>
<td>Krovak Oblique Conformal Conic</td>
<td>2020</td>
</tr>
<tr>
<td>Label</td>
<td>2880</td>
</tr>
<tr>
<td>Label Display Options</td>
<td>1128</td>
</tr>
<tr>
<td>LabelAlignMulti</td>
<td>2552</td>
</tr>
<tr>
<td>LabelAlignX</td>
<td>2553</td>
</tr>
<tr>
<td>LabelAlignY</td>
<td>2554</td>
</tr>
<tr>
<td>Labels</td>
<td>1063</td>
</tr>
<tr>
<td>Labels - Synchronized</td>
<td>3279</td>
</tr>
<tr>
<td>Labels - Text</td>
<td>3278</td>
</tr>
<tr>
<td>Labels - Unlink</td>
<td>3280</td>
</tr>
<tr>
<td>Labels Components - Context Menus</td>
<td>3511</td>
</tr>
<tr>
<td>Labels Object</td>
<td>2883</td>
</tr>
<tr>
<td>LabelSelectionSet</td>
<td>2889</td>
</tr>
<tr>
<td>LabelSet</td>
<td>2890</td>
</tr>
<tr>
<td>LabelWindow</td>
<td>2892</td>
</tr>
<tr>
<td>Lambert Azimuthal Equal Area</td>
<td>1998</td>
</tr>
<tr>
<td>Lambert Conformal Conic</td>
<td>2021</td>
</tr>
<tr>
<td>Lambert Cylindrical Equal Area</td>
<td>2026, 2028</td>
</tr>
<tr>
<td>Landsat</td>
<td>938</td>
</tr>
<tr>
<td>Laplace</td>
<td>988</td>
</tr>
<tr>
<td>Latitude / Longitude Projection</td>
<td>1991</td>
</tr>
<tr>
<td>Layer</td>
<td>2895</td>
</tr>
<tr>
<td>Layer Opacity</td>
<td>1011</td>
</tr>
<tr>
<td>Layer Restrictions</td>
<td>1007</td>
</tr>
<tr>
<td>Layer Tabs</td>
<td>995</td>
</tr>
<tr>
<td>Layers</td>
<td>995, 3185</td>
</tr>
<tr>
<td>Layers and Commands</td>
<td>1001</td>
</tr>
<tr>
<td>LayerSet</td>
<td>2896</td>
</tr>
<tr>
<td>Layout</td>
<td>2897</td>
</tr>
<tr>
<td>Layout Templates</td>
<td>1628</td>
</tr>
<tr>
<td>LayoutBorder</td>
<td>2555</td>
</tr>
<tr>
<td>LayoutEntry</td>
<td>2901</td>
</tr>
<tr>
<td>LayoutEntrySet</td>
<td>2905</td>
</tr>
<tr>
<td>LayoutPaging</td>
<td>2556</td>
</tr>
<tr>
<td>Layouts</td>
<td>1590</td>
</tr>
<tr>
<td>LayoutScope</td>
<td>2557</td>
</tr>
<tr>
<td>LayoutState</td>
<td>2558</td>
</tr>
<tr>
<td>LayoutType</td>
<td>2559</td>
</tr>
<tr>
<td>LayoutWindow</td>
<td>2906</td>
</tr>
<tr>
<td>Term</td>
<td>Page</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>Mosaic</td>
<td>4101</td>
</tr>
<tr>
<td>Motion Blur</td>
<td>3355</td>
</tr>
<tr>
<td>Mouse Moves</td>
<td>37</td>
</tr>
<tr>
<td>Move Horizontally</td>
<td>664</td>
</tr>
<tr>
<td>Move Vertically</td>
<td>664</td>
</tr>
<tr>
<td>MrSID</td>
<td>1776</td>
</tr>
<tr>
<td>Multimedia Control</td>
<td>2489</td>
</tr>
<tr>
<td>MultiNet</td>
<td>1716</td>
</tr>
<tr>
<td>Multipage Layouts</td>
<td>1655</td>
</tr>
<tr>
<td>multipoint</td>
<td>2223</td>
</tr>
<tr>
<td>Multi-User</td>
<td>413</td>
</tr>
<tr>
<td>Multi-User Editing</td>
<td>413</td>
</tr>
<tr>
<td>Multi-User Editing of Linked Drawings</td>
<td>413</td>
</tr>
<tr>
<td>MWS</td>
<td>1743</td>
</tr>
<tr>
<td>My First Map</td>
<td>3635</td>
</tr>
<tr>
<td>N Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>N Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>NADCON</td>
<td>226</td>
</tr>
<tr>
<td>National Grids</td>
<td>1984</td>
</tr>
<tr>
<td>Native Coordinates</td>
<td>1244</td>
</tr>
<tr>
<td>Navigation Toolbar</td>
<td>3475</td>
</tr>
<tr>
<td>Nearest Neighbors</td>
<td>666</td>
</tr>
<tr>
<td>NetCDF</td>
<td>1784</td>
</tr>
<tr>
<td>Network User's Introduction</td>
<td>2067</td>
</tr>
<tr>
<td>Networks</td>
<td>2068</td>
</tr>
<tr>
<td>New Window</td>
<td>37</td>
</tr>
<tr>
<td>Node Points</td>
<td>643</td>
</tr>
<tr>
<td>Noise</td>
<td>3346</td>
</tr>
<tr>
<td>Normalize Metric</td>
<td>667</td>
</tr>
<tr>
<td>Normalize Topology</td>
<td>670</td>
</tr>
<tr>
<td>North Arrow</td>
<td>3145</td>
</tr>
<tr>
<td>NorthArrowType</td>
<td>2567</td>
</tr>
<tr>
<td>Northwind</td>
<td>3633</td>
</tr>
<tr>
<td>Northwood</td>
<td>1794</td>
</tr>
<tr>
<td>Not All Maps are Networks</td>
<td>2072</td>
</tr>
<tr>
<td>Notes</td>
<td>3196</td>
</tr>
<tr>
<td>NTAD</td>
<td>1744</td>
</tr>
<tr>
<td>NTF</td>
<td>1745</td>
</tr>
<tr>
<td>NTv2</td>
<td>226, 2150</td>
</tr>
<tr>
<td>NVIDIA</td>
<td>339</td>
</tr>
<tr>
<td>NVIDIA CUDA</td>
<td>339</td>
</tr>
<tr>
<td>Nwind.mdb</td>
<td>3633</td>
</tr>
<tr>
<td>0</td>
<td>2922</td>
</tr>
<tr>
<td>Object</td>
<td>2922</td>
</tr>
<tr>
<td>Object Coordinates</td>
<td>3508</td>
</tr>
<tr>
<td>Object Fields</td>
<td>3510</td>
</tr>
<tr>
<td>ObjectSet</td>
<td>2925</td>
</tr>
<tr>
<td>ObjectType</td>
<td>2570</td>
</tr>
<tr>
<td>Oblique Azimuthal</td>
<td>1975</td>
</tr>
<tr>
<td>Oblique Cylindrical</td>
<td>1975</td>
</tr>
<tr>
<td>Oblique Mercator</td>
<td>2035</td>
</tr>
<tr>
<td>OGC</td>
<td>292</td>
</tr>
<tr>
<td>Term</td>
<td>Page</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>PointUSNGPrecision</td>
<td>2572</td>
</tr>
<tr>
<td>Polar Azimuthal</td>
<td>1975</td>
</tr>
<tr>
<td>Polar Stereographic</td>
<td>2008</td>
</tr>
<tr>
<td>Polyconic</td>
<td>2024</td>
</tr>
<tr>
<td>Polygons</td>
<td>4316</td>
</tr>
<tr>
<td>POSITION</td>
<td>3625</td>
</tr>
<tr>
<td>POSITION Operator</td>
<td>3625</td>
</tr>
<tr>
<td>Posterize</td>
<td>3358</td>
</tr>
<tr>
<td>PostGIS</td>
<td>312</td>
</tr>
<tr>
<td>PostgreSQL</td>
<td>312</td>
</tr>
<tr>
<td>Precision</td>
<td>3226</td>
</tr>
<tr>
<td>Print</td>
<td>3074</td>
</tr>
<tr>
<td>Printing</td>
<td>1587</td>
</tr>
<tr>
<td>Printing Tables</td>
<td>1673</td>
</tr>
<tr>
<td>Printing to Files or Images</td>
<td>248</td>
</tr>
<tr>
<td>Printing to Scale</td>
<td>1590, 1640</td>
</tr>
<tr>
<td>PRJ</td>
<td>226, 3075, 3080</td>
</tr>
<tr>
<td>Problems when using Windows 95, 98 or Me</td>
<td>4185</td>
</tr>
<tr>
<td>Problems with Commands or Tools</td>
<td>4190</td>
</tr>
<tr>
<td>Problems with Installation or Configuration</td>
<td>4183</td>
</tr>
<tr>
<td>Problems with Performance</td>
<td>4200</td>
</tr>
<tr>
<td>Problems with Projections</td>
<td>4202</td>
</tr>
<tr>
<td>Problems with Tables</td>
<td>4196</td>
</tr>
<tr>
<td>Problems with the Internet Map Server</td>
<td>4204</td>
</tr>
<tr>
<td>Profile</td>
<td>2939</td>
</tr>
<tr>
<td>Profiles and Elevations</td>
<td>1555</td>
</tr>
<tr>
<td>Programming</td>
<td>17</td>
</tr>
<tr>
<td>Programming Manifold</td>
<td>2371</td>
</tr>
<tr>
<td>Programming Reference</td>
<td>2519</td>
</tr>
<tr>
<td>Progress</td>
<td>2943</td>
</tr>
<tr>
<td>Progress Bar</td>
<td>2467</td>
</tr>
<tr>
<td>Project Pane - Context Menus</td>
<td>3496</td>
</tr>
<tr>
<td>Project Pane - Open Data Source</td>
<td>3500</td>
</tr>
<tr>
<td>Project Pane - Paste as Drawing</td>
<td>3501</td>
</tr>
<tr>
<td>Project Pane - Paste as Surface</td>
<td>3502</td>
</tr>
<tr>
<td>Projected US Map</td>
<td>3663</td>
</tr>
<tr>
<td>Projecting a Map</td>
<td>1881</td>
</tr>
<tr>
<td>Projection</td>
<td>3075, 3080</td>
</tr>
<tr>
<td>Projections</td>
<td>1871</td>
</tr>
<tr>
<td>Projections and Images</td>
<td>1917</td>
</tr>
<tr>
<td>Projections and Imported Components</td>
<td>230</td>
</tr>
<tr>
<td>Projections and Legacy Formats</td>
<td>1890</td>
</tr>
<tr>
<td>Projections Quick Reference</td>
<td>1887</td>
</tr>
<tr>
<td>Projections Readings</td>
<td>1963</td>
</tr>
<tr>
<td>Projections Tutorial</td>
<td>1895</td>
</tr>
<tr>
<td>projects</td>
<td>59</td>
</tr>
<tr>
<td>Properties</td>
<td>3223</td>
</tr>
<tr>
<td>PS</td>
<td>1832, 1836</td>
</tr>
<tr>
<td>Pseudocylindrical</td>
<td>1980</td>
</tr>
<tr>
<td>PsPixelEncoding</td>
<td>2573</td>
</tr>
<tr>
<td>Public Access to Public Data</td>
<td>4303</td>
</tr>
<tr>
<td>Publishing Multiple Pages</td>
<td>2113</td>
</tr>
<tr>
<td>PythonScript</td>
<td>2380</td>
</tr>
<tr>
<td>Q</td>
<td></td>
</tr>
<tr>
<td>Quantize</td>
<td>3359</td>
</tr>
<tr>
<td>Queries</td>
<td>1353</td>
</tr>
<tr>
<td>Queries and Images or Surfaces</td>
<td>1390</td>
</tr>
<tr>
<td>Queries Using Multiple Tables</td>
<td>1373</td>
</tr>
<tr>
<td>Query</td>
<td>2944</td>
</tr>
<tr>
<td>Query Templates</td>
<td>4033</td>
</tr>
<tr>
<td>Query Toolbar</td>
<td>3476</td>
</tr>
<tr>
<td>QueryParameter</td>
<td>2948</td>
</tr>
<tr>
<td>QueryParameterSet</td>
<td>2949</td>
</tr>
<tr>
<td>QueryType Constants</td>
<td>2574</td>
</tr>
<tr>
<td>R</td>
<td></td>
</tr>
<tr>
<td>radio button</td>
<td>37</td>
</tr>
<tr>
<td>Rank Columns</td>
<td>1310</td>
</tr>
<tr>
<td>Rank Columns / Decision Support System</td>
<td>4044</td>
</tr>
<tr>
<td>Raster</td>
<td>17</td>
</tr>
<tr>
<td>Raster Extensions</td>
<td>3553</td>
</tr>
<tr>
<td>rasterization</td>
<td>4325</td>
</tr>
<tr>
<td>Rasterize</td>
<td>4325</td>
</tr>
<tr>
<td>Raw Text</td>
<td>1795</td>
</tr>
<tr>
<td>Read-only</td>
<td>37</td>
</tr>
<tr>
<td>RecentDataSource</td>
<td>2950</td>
</tr>
<tr>
<td>RecentDataSourceSet</td>
<td>2951</td>
</tr>
<tr>
<td>Record</td>
<td>2952</td>
</tr>
<tr>
<td>RecordSet</td>
<td>2954</td>
</tr>
<tr>
<td>Rect</td>
<td>2960</td>
</tr>
<tr>
<td>Redo</td>
<td>194</td>
</tr>
<tr>
<td>References</td>
<td>3402</td>
</tr>
<tr>
<td>Refresh / Autorefresh View</td>
<td>3155</td>
</tr>
<tr>
<td>Refresh Data</td>
<td>3154</td>
</tr>
<tr>
<td>Refresh View</td>
<td>3155</td>
</tr>
<tr>
<td>Regular Conic</td>
<td>1975</td>
</tr>
<tr>
<td>Regular Cylindrical</td>
<td>1975</td>
</tr>
<tr>
<td>Regular Expressions</td>
<td>1324</td>
</tr>
<tr>
<td>Relation</td>
<td>2962</td>
</tr>
<tr>
<td>Relations</td>
<td>1268</td>
</tr>
<tr>
<td>RelationSet</td>
<td>2963</td>
</tr>
<tr>
<td>Relative Neighborhood Network</td>
<td>674</td>
</tr>
<tr>
<td>Relief</td>
<td>3362</td>
</tr>
<tr>
<td>Relink</td>
<td>3364, 3397</td>
</tr>
<tr>
<td>Remove Duplicates</td>
<td>677</td>
</tr>
<tr>
<td>Replace</td>
<td>3102</td>
</tr>
<tr>
<td>Repositioning</td>
<td>37</td>
</tr>
<tr>
<td>Repositioning Layers</td>
<td>1019</td>
</tr>
<tr>
<td>Reserved Words</td>
<td>3529</td>
</tr>
<tr>
<td>Resize</td>
<td>3383</td>
</tr>
<tr>
<td>Resizing</td>
<td>37</td>
</tr>
<tr>
<td>restore</td>
<td>37</td>
</tr>
<tr>
<td>Reverse Lines</td>
<td>678</td>
</tr>
<tr>
<td>Review</td>
<td>3199</td>
</tr>
<tr>
<td>RGB</td>
<td>830</td>
</tr>
<tr>
<td>RGB Images</td>
<td>789</td>
</tr>
<tr>
<td>RGBA</td>
<td>830, 859</td>
</tr>
<tr>
<td>Topic</td>
<td>Page</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>RGBA Images</td>
<td>789</td>
</tr>
<tr>
<td>RGBA Pixel Transparency</td>
<td>859</td>
</tr>
<tr>
<td>Rich Text Box</td>
<td>2473</td>
</tr>
<tr>
<td>RichControl</td>
<td>2964</td>
</tr>
<tr>
<td>RichForm</td>
<td>2965</td>
</tr>
<tr>
<td>Rijksdriehoekstelsel</td>
<td>1984</td>
</tr>
<tr>
<td>Robinson</td>
<td>2051</td>
</tr>
<tr>
<td>Rotate</td>
<td>991</td>
</tr>
<tr>
<td>Rotate (Drawings)</td>
<td>680</td>
</tr>
<tr>
<td>Rotating Labels</td>
<td>1139</td>
</tr>
<tr>
<td>Rotating Point Styles</td>
<td>498</td>
</tr>
<tr>
<td>Route</td>
<td>2967</td>
</tr>
<tr>
<td>Runtime Licenses</td>
<td>4258</td>
</tr>
<tr>
<td>Sample Queries</td>
<td>4036</td>
</tr>
<tr>
<td>Satellite View</td>
<td>1980</td>
</tr>
<tr>
<td>Saturation</td>
<td>836</td>
</tr>
<tr>
<td>Save / Save As</td>
<td>3064</td>
</tr>
<tr>
<td>Save control points</td>
<td>1955</td>
</tr>
<tr>
<td>Save, Load Mask/Channel</td>
<td>917, 3099</td>
</tr>
<tr>
<td>saved selections</td>
<td>3208</td>
</tr>
<tr>
<td>Saving Components</td>
<td>77</td>
</tr>
<tr>
<td>Saving Invisible Pixels</td>
<td>838</td>
</tr>
<tr>
<td>Saving the Alpha Channel</td>
<td>859</td>
</tr>
<tr>
<td>Scale Bar</td>
<td>3147</td>
</tr>
<tr>
<td>Scale Horizontally</td>
<td>681</td>
</tr>
<tr>
<td>Scale Vertically</td>
<td>681</td>
</tr>
<tr>
<td>Scope</td>
<td>2575</td>
</tr>
<tr>
<td>Script</td>
<td>2968</td>
</tr>
<tr>
<td>Script - Compile to DLL</td>
<td>3400</td>
</tr>
<tr>
<td>Script - Language</td>
<td>3401</td>
</tr>
<tr>
<td>Script - References</td>
<td>3402</td>
</tr>
<tr>
<td>Script Examples</td>
<td>2393</td>
</tr>
<tr>
<td>Script Menu</td>
<td>3399</td>
</tr>
<tr>
<td>Scripting Reference</td>
<td>2520</td>
</tr>
<tr>
<td>Scripts</td>
<td>2380</td>
</tr>
<tr>
<td>scroll</td>
<td>37</td>
</tr>
<tr>
<td>Scroll Bars</td>
<td>59, 2448</td>
</tr>
<tr>
<td>SDE</td>
<td>318</td>
</tr>
<tr>
<td>SDTS</td>
<td>1746</td>
</tr>
<tr>
<td>Security and Access Control</td>
<td>271</td>
</tr>
<tr>
<td>Segmentization</td>
<td>1929</td>
</tr>
<tr>
<td>Segmentize</td>
<td>582, 1929</td>
</tr>
<tr>
<td>Segments</td>
<td>683</td>
</tr>
<tr>
<td>Select Add</td>
<td>101</td>
</tr>
<tr>
<td>Select Adjacent to</td>
<td>684</td>
</tr>
<tr>
<td>Select All</td>
<td>3093</td>
</tr>
<tr>
<td>Select All / None / Inverse</td>
<td>3093</td>
</tr>
<tr>
<td>Select Areas</td>
<td>101</td>
</tr>
<tr>
<td>Select Box</td>
<td>101</td>
</tr>
<tr>
<td>Select Box on Center</td>
<td>101</td>
</tr>
<tr>
<td>Select Circle</td>
<td>101</td>
</tr>
<tr>
<td>Select Circle on Center</td>
<td>101</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Contained / Containing</td>
<td>685</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Contained / Containing</td>
<td>685</td>
</tr>
<tr>
<td>Select Contained / Containing</td>
<td>685</td>
</tr>
<tr>
<td>Select Contained / Containing</td>
<td>685</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Select Critical Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Index</td>
<td></td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>---</td>
</tr>
<tr>
<td>Server Console</td>
<td>3299</td>
</tr>
<tr>
<td>Service Centers</td>
<td>693</td>
</tr>
<tr>
<td>Shape Hull</td>
<td>697</td>
</tr>
<tr>
<td>Shapefiles</td>
<td>1748, 1830</td>
</tr>
<tr>
<td>Shared Edit</td>
<td>566</td>
</tr>
<tr>
<td>Sharpen</td>
<td>992</td>
</tr>
<tr>
<td>Shield</td>
<td>1134</td>
</tr>
<tr>
<td>Shortest Path</td>
<td>691</td>
</tr>
<tr>
<td>Shortest Path over Land</td>
<td>4051</td>
</tr>
<tr>
<td>SHP</td>
<td>1748, 1830</td>
</tr>
<tr>
<td>SID</td>
<td>1776</td>
</tr>
<tr>
<td>Simple Queries</td>
<td>1371</td>
</tr>
<tr>
<td>Simplify</td>
<td>584, 3366</td>
</tr>
<tr>
<td>Sinusoidal</td>
<td>2053</td>
</tr>
<tr>
<td>SKIP</td>
<td>3591</td>
</tr>
<tr>
<td>Slider</td>
<td>2480</td>
</tr>
<tr>
<td>slope</td>
<td>1541, 3099</td>
</tr>
<tr>
<td>Slope and Aspect from Elevation Data</td>
<td>4326</td>
</tr>
<tr>
<td>Smart Mouse</td>
<td>129</td>
</tr>
<tr>
<td>Smart Mouse Selection</td>
<td>129</td>
</tr>
<tr>
<td>Smooth</td>
<td>897</td>
</tr>
<tr>
<td>Snap To</td>
<td>3111</td>
</tr>
<tr>
<td>Snap to Surface</td>
<td>3439</td>
</tr>
<tr>
<td>Snap Toolbar</td>
<td>3484</td>
</tr>
<tr>
<td>SOME</td>
<td>3589</td>
</tr>
<tr>
<td>Sort</td>
<td>3132</td>
</tr>
<tr>
<td>Space Oblique Mercator</td>
<td>2060</td>
</tr>
<tr>
<td>Space Oblique Mercator for Landsat</td>
<td>2060</td>
</tr>
<tr>
<td>Spanning Tree</td>
<td>698</td>
</tr>
<tr>
<td>Spatial</td>
<td>17</td>
</tr>
<tr>
<td>Spatial DBMS</td>
<td>312, 2223</td>
</tr>
<tr>
<td>Spatial DBMS Facilities</td>
<td>2223</td>
</tr>
<tr>
<td>Spatial Extensions</td>
<td>3544</td>
</tr>
<tr>
<td>Spatial Geocoding with Match</td>
<td>1436</td>
</tr>
<tr>
<td>spatial index</td>
<td>2223</td>
</tr>
<tr>
<td>Spatial Overlay</td>
<td>586, 3261</td>
</tr>
<tr>
<td>spatial queries</td>
<td>4042</td>
</tr>
<tr>
<td>Specifying Views in Layouts</td>
<td>1640</td>
</tr>
<tr>
<td>spin buttons</td>
<td>2484</td>
</tr>
<tr>
<td>Spin buttons</td>
<td>37</td>
</tr>
<tr>
<td>Spline</td>
<td>701</td>
</tr>
<tr>
<td>SPLIT BY</td>
<td>3626</td>
</tr>
<tr>
<td>SPLIT BY Clause</td>
<td>3626</td>
</tr>
<tr>
<td>Split With</td>
<td>703</td>
</tr>
<tr>
<td>SQL Reserved Words</td>
<td>3529</td>
</tr>
<tr>
<td>SQL Reserved Words / Index</td>
<td>3529</td>
</tr>
<tr>
<td>SQL Server</td>
<td>321, 2262, 3815</td>
</tr>
<tr>
<td>SQL Server 2005</td>
<td>312</td>
</tr>
<tr>
<td>SQL Server 2008 312, 2262, 2268, 2278, 2300, 2331</td>
<td>358</td>
</tr>
<tr>
<td>SQL Server Express Edition</td>
<td>2262</td>
</tr>
<tr>
<td>SQL Server Spatial DBMS Facilities</td>
<td>1798</td>
</tr>
<tr>
<td>SRTM</td>
<td>1798</td>
</tr>
<tr>
<td>Standard Palettes</td>
<td>865</td>
</tr>
<tr>
<td>Standardize</td>
<td>3390</td>
</tr>
<tr>
<td>State Plane Coordinate System</td>
<td>1990</td>
</tr>
<tr>
<td>Static Text Box</td>
<td>2439</td>
</tr>
<tr>
<td>Statistics</td>
<td>17</td>
</tr>
<tr>
<td>Storing an Image in SQL Server 2008</td>
<td>2331</td>
</tr>
<tr>
<td>Storing Drawings in SQL Server</td>
<td>3815</td>
</tr>
<tr>
<td>Street Address Geocoding</td>
<td>1407</td>
</tr>
<tr>
<td>SUBSTRING</td>
<td>3628</td>
</tr>
<tr>
<td>SUBSTRING Operator</td>
<td>3628</td>
</tr>
<tr>
<td>SubtypeAny</td>
<td>2577</td>
</tr>
<tr>
<td>SubtypeDb</td>
<td>2578</td>
</tr>
<tr>
<td>SubtypeDbf</td>
<td>2579</td>
</tr>
<tr>
<td>SubtypeMdb</td>
<td>2580</td>
</tr>
<tr>
<td>SubtypeTaif</td>
<td>2581</td>
</tr>
<tr>
<td>SubtypeWk</td>
<td>2582</td>
</tr>
<tr>
<td>SubtypeXls</td>
<td>2583</td>
</tr>
<tr>
<td>Surface</td>
<td>17, 2971</td>
</tr>
<tr>
<td>Surface - Convert To</td>
<td>3416</td>
</tr>
<tr>
<td>Surface - Invert</td>
<td>3404</td>
</tr>
<tr>
<td>Surface - Noise</td>
<td>3410</td>
</tr>
<tr>
<td>Surface - Posterize</td>
<td>3407</td>
</tr>
<tr>
<td>Surface - Quantize</td>
<td>3414</td>
</tr>
<tr>
<td>Surface - Resize</td>
<td>3415</td>
</tr>
<tr>
<td>Surface - Threshold</td>
<td>3405</td>
</tr>
<tr>
<td>Surface - Tile</td>
<td>3411</td>
</tr>
<tr>
<td>Surface - Transform</td>
<td>3412</td>
</tr>
<tr>
<td>Surface Display Options</td>
<td>1470</td>
</tr>
<tr>
<td>Surface Menu</td>
<td>3402</td>
</tr>
<tr>
<td>Surface Tools</td>
<td>1527</td>
</tr>
<tr>
<td>Surface Window Menus and Controls</td>
<td>1466</td>
</tr>
<tr>
<td>Surfaces and Terrains</td>
<td>1461</td>
</tr>
<tr>
<td>SurfaceSelectionSet</td>
<td>2976</td>
</tr>
<tr>
<td>Synchronized</td>
<td>3279</td>
</tr>
<tr>
<td>Synchronized Labels</td>
<td>1092</td>
</tr>
<tr>
<td>System Activity Indicator</td>
<td>59</td>
</tr>
<tr>
<td>System Info Control</td>
<td>2503</td>
</tr>
<tr>
<td>Tab Strip</td>
<td>2486</td>
</tr>
<tr>
<td>Table</td>
<td>2977</td>
</tr>
<tr>
<td>Table - Add Active Column</td>
<td>3386</td>
</tr>
<tr>
<td>Table - Add Rank Column</td>
<td>3387</td>
</tr>
<tr>
<td>Table - Address - Geocode</td>
<td>3388</td>
</tr>
<tr>
<td>Table - Address - Standardize</td>
<td>3390</td>
</tr>
<tr>
<td>Table - Design</td>
<td>3391</td>
</tr>
<tr>
<td>Table - Match</td>
<td>3398</td>
</tr>
<tr>
<td>Table - Relations</td>
<td>3393</td>
</tr>
<tr>
<td>Table - Relink / Unlink</td>
<td>3397</td>
</tr>
<tr>
<td>Table - Transform Toolbar and Operators</td>
<td>4327</td>
</tr>
<tr>
<td>Table Window Menus and Controls</td>
<td>1233</td>
</tr>
<tr>
<td>Tables</td>
<td>1221</td>
</tr>
<tr>
<td>Tables - Context Menus</td>
<td>3521</td>
</tr>
<tr>
<td>Term</td>
<td>Page</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>TableSelectionSet</td>
<td>2981</td>
</tr>
<tr>
<td>TableWindow</td>
<td>2982</td>
</tr>
<tr>
<td>TAIF</td>
<td>1757</td>
</tr>
<tr>
<td>Technical Support</td>
<td>5, 4213</td>
</tr>
<tr>
<td>TeleAtlas</td>
<td>1738</td>
</tr>
<tr>
<td>TEMP</td>
<td>4181</td>
</tr>
<tr>
<td>Template</td>
<td>3127</td>
</tr>
<tr>
<td>Templates</td>
<td>1628</td>
</tr>
<tr>
<td>Temporary Files</td>
<td>4181</td>
</tr>
<tr>
<td>Terminology in GIS</td>
<td>4284</td>
</tr>
<tr>
<td>Terrain</td>
<td>17, 2983</td>
</tr>
<tr>
<td>Terrain - Clouds</td>
<td>3433</td>
</tr>
<tr>
<td>Terrain - Fog</td>
<td>3434</td>
</tr>
<tr>
<td>Terrain - Lighting</td>
<td>3435</td>
</tr>
<tr>
<td>Terrain - Overlay</td>
<td>3426</td>
</tr>
<tr>
<td>Terrain - Snap to Surface</td>
<td>3439</td>
</tr>
<tr>
<td>Terrain - Surface</td>
<td>3421</td>
</tr>
<tr>
<td>Terrain - Water</td>
<td>3437</td>
</tr>
<tr>
<td>Terrains</td>
<td>1483</td>
</tr>
<tr>
<td>TerrainWindow</td>
<td>2988</td>
</tr>
<tr>
<td>TerraServer</td>
<td>893</td>
</tr>
<tr>
<td>Text Box</td>
<td>2436</td>
</tr>
<tr>
<td>TextControl</td>
<td>3058</td>
</tr>
<tr>
<td>TGA</td>
<td>1832</td>
</tr>
<tr>
<td>The Data Source Dialog</td>
<td>320</td>
</tr>
<tr>
<td>The Earth as an Ellipsoid</td>
<td>1968</td>
</tr>
<tr>
<td>Thematic Formatting</td>
<td>474</td>
</tr>
<tr>
<td>Thematic Formatting and Labels</td>
<td>1118</td>
</tr>
<tr>
<td>Thematic Formatting Example</td>
<td>494</td>
</tr>
<tr>
<td>Thematic Mapping</td>
<td>4319</td>
</tr>
<tr>
<td>Theme</td>
<td>2990</td>
</tr>
<tr>
<td>Themes</td>
<td>502</td>
</tr>
<tr>
<td>Thiessen</td>
<td>710</td>
</tr>
<tr>
<td>Thiessen Polygons</td>
<td>4328</td>
</tr>
<tr>
<td>Threshold</td>
<td>3368</td>
</tr>
<tr>
<td>Threshold Color</td>
<td>3375</td>
</tr>
<tr>
<td>ThunderScan</td>
<td>1841</td>
</tr>
<tr>
<td>tif 1832</td>
<td></td>
</tr>
<tr>
<td>TriffCompression</td>
<td>2584</td>
</tr>
<tr>
<td>TIGER/Line</td>
<td>1758</td>
</tr>
<tr>
<td>Tile</td>
<td>3381</td>
</tr>
<tr>
<td>Tile Horizontally</td>
<td>37</td>
</tr>
<tr>
<td>Tile Vertically</td>
<td>37</td>
</tr>
<tr>
<td>Tilted Perspective</td>
<td>2010</td>
</tr>
<tr>
<td>TIN</td>
<td>1789</td>
</tr>
<tr>
<td>Tokens</td>
<td>1351</td>
</tr>
<tr>
<td>Tolerance Setting</td>
<td>101</td>
</tr>
<tr>
<td>Tool Bar</td>
<td>2493</td>
</tr>
<tr>
<td>Tool Properties</td>
<td>3211</td>
</tr>
<tr>
<td>Toolbars</td>
<td>37, 3454</td>
</tr>
<tr>
<td>Tools - Add-In Manager</td>
<td>3280</td>
</tr>
<tr>
<td>Tools - Administrator Console</td>
<td>3282</td>
</tr>
<tr>
<td>Tools - Batch Export</td>
<td>3301</td>
</tr>
</tbody>
</table>
Index

Transform - Inner Buffers 641
Transform - Intersect Lines 663
Transform - Intersection Points 642
Transform - Join Lines 640
Transform - Laplace 988
Transform - Median Cross / Median Square 989
Transform - Move Horizontally / Move Vertically 664
Transform - Nearest / Farthest Neighbors 666
Transform - Node Points 643
Transform - Normalize Metric 667
Transform - Normalize Topology 670
Transform - Points 645
Transform - Relative Neighborhood Network 674
Transform - Remove Duplicates 677
Transform - Reverse Lines 678
Transform - Rotate 991
Transform - Rotate (Drawings) 680
Transform - Scale / Scale Horizontally / Scale Vertically 681
Transform - Segmentize 683
Transform - Select Adjacent to 684
Transform - Select Contained / Containing 685
Transform - Select Euclidean Point Coverage 687
Transform - Select Intersecting 690
Transform - Select Service Centers 693
Transform - Select Shortest Path 691
Transform - Select Touching 696
Transform - Shape Hull 697
Transform - Sharpen 992
Transform - Span Excluding / Including 1352
Transform - Spanning Tree 698
Transform - Spline 701
Transform - Split With 703
Transform - Surface - Interpolate Operators 1523
Transform - Surface - Tile Operators 1526
Transform - Triangulation 705
Transform - Union 709
Transform - Using Tokens and Text Strings 1351
Transform - Voronoi Operators 710
Transform Dialog Functions and Operators 1541
Transform Operators - Drawings 605
Transform Operators - Images 971
Transform Operators - Surfaces 1520
Transform Operators - Tables 1335
TRANSFORM Statement 3586
Transform Toolbar 3495
Transform Toolbar - Drawings 601
Transform Toolbar - Images 969
Transform Toolbar - Surfaces 1518
Transform Toolbar - Tables 1332
Transforming Surfaces 1530
Transparency 859
Transparent Area Styles 454
Transparent Color 442
Transverse Cylindrical 1975
Transverse Mercator 2037
Tree View 2482
TriangleSet 2997
Triangulation 705
TRIM 3629
TRIM Operator 3629
Troubleshooting 4181
Turning Layers Off/On by Zoom 3752
TurnStyle 2587
U
Undo 194
Undo / Redo 3089
undock 37
Undocking 37
Union 593, 709
UNION 3630
UNION Operator 3630
UNIQUE 3631
UNIQUE Operator 3631
Unit 2998
Units 4279
UnitSet 2999
Universal Polar Stereographic 2061
Universal Transverse Mercator (UTM) 2062
Unknown Errors 4200
Unlink 3280, 3364, 3397
Update 593
Update Batching 2389
UPDATE Statement 3588
Updates 3449
UpDown 37, 2484
US Map 3663
User Interface Design 4282
User Interface Scripting 2391
UserInterface 3000
UserInterfaceControl 3002
UserInterfaceControlItem 3003
UserInterfaceControlItemSet 3004
UserInterfaceControlSet 3005
UserInterfaceDialog 3006
UserInterfaceDialogSet 3007
Using Administrator Console 2202
Using Cut, Copy and Paste in Drawings 519
Using Layers 3677
Using Projections 226
Using RAM and other Machine Resources 4300
Using SQL to Select Map Objects 4042
Using Surfaces as Images 1479
Using the Manifold ODBC Driver 292
UTM 2062
<table>
<thead>
<tr>
<th>V</th>
<th>ViewSet</th>
</tr>
</thead>
<tbody>
<tr>
<td>ValueType</td>
<td>viewshed</td>
</tr>
<tr>
<td>Van der Grinten</td>
<td>Virtual Earth</td>
</tr>
<tr>
<td>Van der Grinten IV</td>
<td>Virtual Tables</td>
</tr>
<tr>
<td>VB .NET</td>
<td>Virtual Tables for Images and Surfaces</td>
</tr>
<tr>
<td>VBScript</td>
<td>visibility</td>
</tr>
<tr>
<td>VCT</td>
<td>visibility zones</td>
</tr>
<tr>
<td>vectorization</td>
<td>Visible Area</td>
</tr>
<tr>
<td>Vectorizing</td>
<td>Voronoi Areas</td>
</tr>
<tr>
<td>Version</td>
<td>Voronoi Diagram</td>
</tr>
<tr>
<td>Vertical Mapper</td>
<td>Voronoi Lines</td>
</tr>
<tr>
<td>Vertical Perspective</td>
<td>Voronoi Points</td>
</tr>
<tr>
<td>Vertical Scroll bar</td>
<td>VScrollControl</td>
</tr>
<tr>
<td>View</td>
<td></td>
</tr>
<tr>
<td>View - Columns</td>
<td>WAAS</td>
</tr>
<tr>
<td>View - Display Options</td>
<td>Wagner IV</td>
</tr>
<tr>
<td>View - Full Screen</td>
<td>Wagner VII</td>
</tr>
<tr>
<td>View - Graticule</td>
<td>Warping Images</td>
</tr>
<tr>
<td>View - Grid</td>
<td>Watersheds</td>
</tr>
<tr>
<td>View - Legend</td>
<td>Web</td>
</tr>
<tr>
<td>View - North Arrow</td>
<td>Web Queries</td>
</tr>
<tr>
<td>View - Panes</td>
<td>weights</td>
</tr>
<tr>
<td>View - Panes - Call Stack</td>
<td>Welcome</td>
</tr>
<tr>
<td>View - Panes - Control Points</td>
<td>What about Ajax?</td>
</tr>
<tr>
<td>View - Panes - Errors</td>
<td>What's New</td>
</tr>
<tr>
<td>View - Panes - GPS Console</td>
<td>WHERE Clause</td>
</tr>
<tr>
<td>View - Panes - History</td>
<td>Where to Get Maps</td>
</tr>
<tr>
<td>View - Panes - Info</td>
<td>Why is Clip Coordinates so Slow</td>
</tr>
<tr>
<td>View - Panes - Layers</td>
<td>Window</td>
</tr>
<tr>
<td>View - Panes - Notes</td>
<td>Window - Windows Dialog</td>
</tr>
<tr>
<td>View - Panes - Project</td>
<td>Window Menu</td>
</tr>
<tr>
<td>View - Panes - Review</td>
<td>WindowAlign</td>
</tr>
<tr>
<td>View - Panes - Selections</td>
<td>Windows</td>
</tr>
<tr>
<td>View - Panes - Tool Properties</td>
<td>Windows '95</td>
</tr>
<tr>
<td>View - Panes - Variables</td>
<td>Windows Control Panel</td>
</tr>
<tr>
<td>View - Panes - ViewBots</td>
<td>Windows Dialog</td>
</tr>
<tr>
<td>View - Panes - Views</td>
<td>Windows XP</td>
</tr>
<tr>
<td>View - Panes - Watches</td>
<td>WindowSet</td>
</tr>
<tr>
<td>View - Panes - World</td>
<td>Winkel Tripel</td>
</tr>
<tr>
<td>View - Properties</td>
<td>WMF</td>
</tr>
<tr>
<td>View - Properties - Link / Share Status</td>
<td>WMS</td>
</tr>
<tr>
<td>View - Properties - Precision</td>
<td>Working with Enterprise Edition</td>
</tr>
<tr>
<td>View - Properties - Zooms</td>
<td>Working with GPS Receivers</td>
</tr>
<tr>
<td>View - Refresh / Autorefresh View</td>
<td>Working with Maps</td>
</tr>
<tr>
<td>View - Refresh Data</td>
<td>Working with Queries</td>
</tr>
<tr>
<td>View - Scale Bar</td>
<td>Working with Tables</td>
</tr>
<tr>
<td>View - Selection Filter</td>
<td>Working with Text Components</td>
</tr>
<tr>
<td>View - Sort</td>
<td>World</td>
</tr>
<tr>
<td>View - Structure</td>
<td>World pane</td>
</tr>
<tr>
<td>View - Zoom</td>
<td>X</td>
</tr>
<tr>
<td>ViewBot Operators</td>
<td>XML</td>
</tr>
<tr>
<td>ViewBots</td>
<td>XML Accessory File Format</td>
</tr>
<tr>
<td>Views</td>
<td>XML Files</td>
</tr>
</tbody>
</table>

4366
<table>
<thead>
<tr>
<th>Term</th>
<th>Page 1</th>
<th>Page 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zip Codes</td>
<td>4310</td>
<td></td>
</tr>
<tr>
<td>Zip Codes are Not Areas</td>
<td>4310</td>
<td></td>
</tr>
<tr>
<td>Zone</td>
<td>3014</td>
<td></td>
</tr>
<tr>
<td>Zones</td>
<td>3015</td>
<td></td>
</tr>
<tr>
<td>ZoneSelectionSet</td>
<td>3017</td>
<td></td>
</tr>
<tr>
<td>ZoneSet</td>
<td>3018</td>
<td></td>
</tr>
<tr>
<td>Zoom</td>
<td>3135</td>
<td></td>
</tr>
<tr>
<td>Zoom Box</td>
<td></td>
<td>3475</td>
</tr>
<tr>
<td>Zoom In</td>
<td></td>
<td>3135, 3475</td>
</tr>
<tr>
<td>Zoom Out</td>
<td></td>
<td>3135, 3475</td>
</tr>
<tr>
<td>Zoom Ranges</td>
<td></td>
<td>1023</td>
</tr>
<tr>
<td>Zoom To</td>
<td></td>
<td>3135</td>
</tr>
<tr>
<td>Zoom To Fit</td>
<td></td>
<td>3135, 3475</td>
</tr>
<tr>
<td>Zooms</td>
<td></td>
<td>3228</td>
</tr>
</tbody>
</table>